Qualitative Spatial and Temporal Representation and Reasoning: Efficiency in Time and Space

by

Zhiguo Long

supervised by

Prof. Sanjiang Li

for the degree of

Doctor of Philosophy

Centre for Quantum Software and Information

Faculty of Engineering and Information Technology

University of Technology Sydney (UTS)

January, 2017

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student

_____

ACKNOWLEDGEMENT

First of all, I would like to gratefully thank my supervisor, Prof. Sanjiang Li, for his patient guidance, inspiring discussions, and firm support in financial and research matters, during my four years' PhD candidature. His insistence on mathematical accuracy and emphasis on independence of research have influenced me very much.

I would also like to thank Dr. Steven Schockaert, who has given me many invaluable suggestions and inspirations on research. He is also a great friend in life who is always there to listen and to help. I have learned a lot from his enthusiasm for finding solutions to problems.

Prof. Matt Duckham is another person to whom I would like to express my sincere special thank. He always did his best to help me, without any complaint.

I would like to thank Prof. Jean-François Condotta from Artois University, France, for his kind and diligent review of an early version of this thesis.

In the early stage of my PhD candidature, Dr. Weiming Liu and Dr. Hua Meng provided me with the most patient help in research and life. From them, I learned how to think mathematically, not just to calculate mathematically.

My special thanks also belong to Michael Sioutis and Dr. Jae Hee Lee, for being great collaborators in research and friends in life. Every time I talk with them I will learn interesting things.

I am very grateful to have my family, especially my parents and my wife

supporting me all these years. They are always the source of joy, understanding, and comfort.

I also want to thank Prof. Hui Kou, who recommended me to study under the supervision of Prof. Sanjiang Li at the University of Technology Sydney.

Finally, I want to thank all my friends in Australia, China, and the UK, and colleagues at QCIS, UTS. The friendship with them is one of the most valuable things in my life. With them, I had wonderful memory of playing soccer, tasting delicious food, travelling around, and many more. Among them, I would like to specially thank Dr. Shoaib Jameel for his intriguing introduction of Linux to me and Prof. Xueying Zhang for her advice on life.

TABLE OF CONTENTS

LIST OF TERMS

LIST OF NOTATIONS

a°	Interior of a point set a	21
$\overline{a}$	Closure of a point set a	21
$a_1 R a_2$, $(a_1, a_2) \in R$	(a_1, a_2) satisfies the relation R	16
$\mathsf{adj}(v)$	Adjacency set of vertex v in a graph	28
α, α_i, β, β_j	some basic relations	18
$\alpha_{i_1} \cup \ldots \cup \alpha_{i_k}$	A relation that is the union of some basic relations	18
$\alpha \diamond \beta$, $R \diamond S$	Weak composition of two relations	31
$\alpha \otimes \beta$, $R \otimes S$	Cartesian product of two relations	19
$\mathcal{R} \otimes \mathcal{S}$	Cartesian product of two subclasses	20
$\alpha \in R$	A basic relation α is contained in a relation R	18
$S \subseteq R$	A relation S is contained in a relation R	18
BA(n,m)	Barabási-Albert model with preferential attachment value m and n vertices	89
$\mathsf{id}_{\mathcal{U}}$	Identity relation	16
$\mathsf{B}_{\mathcal{M}}$	The set of basic relation in a qualitative calculus $\mathcal{M}$	17
$\mathcal{C}$	A set of constraint	25
c_i	A constraint	69
$\bar{d}$	Average intersection degree	133
$\delta(a, b)$	A CDC relation	23

F_k	$\{v_j \in \mathsf{adj}(v_k) : j > k\}$	29
$G = (V, E)$	An undirected graph, with vertices V and edges E	27
$G_{\mathcal{N}}$	Constraint graph	27
$\mathcal{M}$	A qualitative calculus	17
$\mathsf{mbr}(a)$	The MBR of a	21
$\mathcal{N}$, $\mathcal{N}'$	Qualitative constraint network	25
$\mathcal{N}_{\mathrm{c}}$	Core of $\mathcal{N}$	103
$\mathcal{N}_{\mathrm{m}}$	Minimal subnetwork of $\mathcal{N}$	42
$\mathcal{N}_{\mathrm{p}}$	A-closure of $\mathcal{N}$	70
$\mathcal{N}_{\mathrm{p}}^{G}$	partially path consistent subnetwork of $\mathcal{N}$ w.r.t. G	35
$\mathcal{N}\|_{V_0}$	The restriction of $\mathcal{N}$ on $V_0 \subseteq V$	27
$\mathcal{N} \models (uRv)$	$\mathcal{N}$ entails (uRv)	47
o_i	Spatial object or region	122
$\mathbf{O}_5$, $\mathbf{O}_8$	Specific sets of relations in RCC5/8	73
π, $(c_1, \ldots, c_s)$	A path in a QCN	69
$\pi_{<i}$, $\pi_{>i}$	$(c_1, \ldots, c_{i-1})$ and $(c_{i+1}, \ldots, c_s)$	69
$\mathrm{CT}(\pi)$	The composition of a path π in a QCN	70
$\|\pi\|$	The length of a path π in a QCN	69
$\mathcal{P}_{xy}^{\mathcal{N}}$	The set of all paths from x to y in a QCN $\mathcal{N}$	70
R, S, T	A relation	16
R_{ij}, S_{ij}, T_{ij}	A relation between v_i and v_j	16
R^{-1}, S^{-1}, T^{-1}	Converse of a relation	16
$\mathcal{R}, \mathcal{S}, \mathcal{T}$, $\mathcal{X}$	Subclass of relations or subalgebra	18
$\widehat{\mathcal{R}}$, $\widehat{\mathcal{S}}$, $\widehat{\mathcal{T}}$, $\widehat{\mathcal{X}}$	Closure of a subclass of relations	60

$\mathbf{Rel}(\mathcal{U})$	The power set of $\mathcal{U} \times \mathcal{U}$	17
σ	A solution of a network	37
t, t_j	Spatial clustering index tile	122
$\mathcal{U}$	Universe, domain	16
(uRv)	A constraint	25
$\star$	Universal relation	16
V	A set of variables or vertices	25
v, w, u	Variable, vertex	25
W	The intersection of the weak compositions of all paths from x to y in $\mathcal{N} \setminus \{(xRy)\}$	74
x_a^-, y_a^-, x_a^+, y_a^+	Lower (upper) bound of the projection of point set a on x/y-axis	21
$I_x(a)$, $[x_a^-, x_a^+]$, $I_y(a)$, $[y_a^-, y_a^+]$	The projection of point set a on x/y-axis	21
$\mathcal{C}_{\mathrm{PA}}$, $\mathcal{S}_{\mathrm{PA}}$	Maximal distributive subalgebras of PA	207
$\mathcal{C}_{\mathrm{IA}}$, $\mathcal{S}_{\mathrm{IA}}$	Maximal distributive subalgebras of IA	208
$\mathcal{D}_{14}^5$, $\mathcal{D}_{20}^5$	Maximal distributive subalgebras of RCC5	210
$\mathcal{D}_{41}^8$, $\mathcal{D}_{64}^8$	Maximal distributive subalgebras of RCC8	210

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ABSTRACT

Qualitative Spatial and Temporal Reasoning (QSTR) provides a human-friendly abstract way to describe and to interpret spatial and temporal information. To describe the qualitative information, QSTR makes use of qualitative relations between entities and usually stores them in a qualitative constraint network (QCN). The QCNs are then used as the basis to process qualitative spatial and temporal information, including qualitative reasoning and query answering.

Time efficiency of reasoning techniques in QSTR is critical for applications to deal with qualitative spatial and temporal information in large-scale datasets. In this thesis, we present a special family of tractable subclasses of relations, called distributive subalgebras. We show that several efficient algorithms are applicable to the QCNs over distributive subalgebras for solving important reasoning problems. We also identify maximal distributive subalgebras for popular relation models in QSTR and point out their connections with several previously identified important subclasses.

Regarding the network representation in QSTR, there are two important problems, which in turn affect the time efficiency of other applications.

First, the network representation can have redundant relations, which will significantly increase the efforts needed for tasks whose efficiency is strongly related to the number of relations in a network. Fortunately, for any QCN over distributive subalgebras of qualitative calculi PA, RCC5, and RCC8, we show that essentially it has a unique subset consisting of non-redundant relations,

which expresses the same qualitative information as the original QCN. We also devise an efficient algorithm to construct such subsets.

Second, the network representation sometimes requires a large storage space when encoding large-scale data. This could severely limit the ability of relation retrieval for any two given spatial entities. In fact, when the size of a QCN becomes large, it might be too costly or even infeasible to fit the QCN into fast accessible storage and relation retrieval will become inefficient. We propose two alternative representation techniques to compactly encode qualitative spatial relations between regions. For this purpose, the first technique uses minimum bounding rectangles (MBRs) to encode both topological relations and directional relations, while the second technique focuses on encoding topological relations by generating axis-aligned rectangles for spatial entities. We show that for large real-world datasets of regions, these two techniques can significantly reduce the storage size of qualitative spatial information and in the meantime the relations between regions can be efficiently inferred from those simple geometric shapes.

Chapter 1

Introduction

Space and time are two of the most important aspects of the universe. In macroscopic scale, space corresponds to our physical world, including the Earth, continents, countries, cities, and so on; time measures our dynamic world, including the past, the present, and the future, as well as the duration of events. In particular, human beings are concerned with these two aspects of the universe in daily life.

In ancient times, understanding space and time has a great impact on survival. For example, it is critical for tribes, towns, and countries to find more fertile lands or more fruitful woods, to know when flood would come, and to identify a better habitat to live. Mastering these requires proficiency in spatial and temporal information. Such information involves various entities such as habitats of plants and animals, water bodies, and surrounding environments of different places.

In modern world, with the development of information technology and artificial intelligence, spatial and temporal information has become even more important. It is rooted in our daily activities such as navigation and planning. Currently, there are two main approaches for handling spatial and temporal

information. The first one deals with the information in a *quantitative* way, by making use of precise numeric models and calculations. The other one processes the information *qualitatively*, by representing and making inferences with non-metric languages.

1.1 Qualitative Spatial and Temporal Information

The quantitative approach of processing spatial and temporal information plays a predominant role in areas such as geographical information science [130], computational geometry [35] and computer vision [27]. Nevertheless, the qualitative approach is still desirable and has promising potential to improve the current technologies concerning spatial and temporal information. For instance, the qualitative approach can improve user experience of the *intelligent personal assistants* (IPAs), such as Apple Siri, Google Now and Microsoft Cortana. These IPAs have attracted more and more attention recently. It is very common that someone asks a question about a place and expects to get a response from the IPA that contains a qualitative description about the place. For example, when the user asks "where is Darling Harbour", the following description about Darling Harbour from Wikipedia[1] would be a desirable response.

> Darling Harbour is a harbour **adjacent to** the city centre of Sydney, New South Wales, Australia. It is also a large recreational and pedestrian precinct that is situated **on western** outskirts of the Sydney central business district. Originally Long Cove, the locality **extends northwards from** Chinatown, **along** both sides of Cockle Bay to Darling Harbour wharf **on the east**, and to the suburb of

[1] `https://en.wikipedia.org/wiki/Darling_Harbour`

Pyrmont **on the west**.

The qualitative relations (bold texts) in the description, such as “adjacent to” and “on the west”, help people to easily understand the connections between spatial entities. Such a description about the spatial information improves the user experience of applications by making the user feel like talking with a human being rather than with a machine. Unfortunately, currently applications such as IPAs have a poor support for qualitative information. For example, instead of a response in the above form, an IPA might only return a map containing Darling Harbour as a point (see Figure 1.1), which barely conveys any useful spatial information for the user.[2]

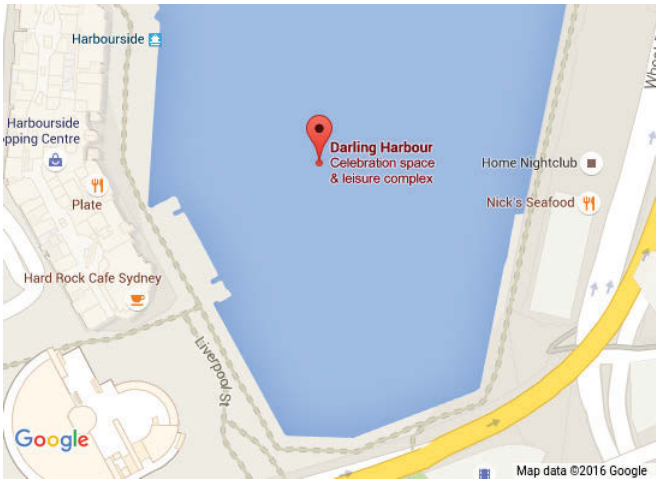


Figure 1.1: Illustration of Darling Harbour on Google Map.

Similar problems also appear in many other applications due to the absence of qualitative information. The following example identifies such a problem in our daily life.

[2]To our knowledge, the IPA Google Now sometimes also returns qualitative descriptions retrieved from Wikipedia, which in turn reflects the importance of integrating qualitative information in such applications.

Example 1.1 (School Catchment Areas)**.** Usually, each public school has a catchment area[3]. Students can register at a school if their neighbourhoods are **contained** in the catchment area of the school. Parents would be interested in whether their children can register at a specific school when they live in a certain neighbourhood. Usually, this information is not explicitly available. In fact, the school catchment areas might be scattered on websites of schools in the form of text descriptions or regions on a map which could be ambiguous, and sometimes the areas are not specified at all. As a result, currently people need to take much effort to find out the containment relation between their neighbourhoods and school catchment areas. It will be much more convenient if we have the qualitative relations explicitly available as a complement of a conventional map, so that people can directly ask if their neighbourhood is contained in a school catchment area.

Therefore, a natural question is how we can make applications to better handle qualitative spatial and temporal information. One promising solution is to build a comprehensive knowledge base that contains tools to process such qualitative information. The knowledge base can then act as an interface to make it much easier for applications to handle qualitative information, which however have several challenges. One challenge is about which qualitative model to use for the knowledge base to better capture human cognition, i.e. the way people use to understand the physical world. Currently this is not clear, although some researches (e.g. [67, 68, 88, 103]) have shown evidence to that qualitative models could be akin to human cognition. In this thesis, we focus on the challenge of how to handle qualitative information abstracted by using existing models, rather than cognition. For example, to build a knowledge base, we need to organise qualitative spatial and temporal information

[3] `http://www.schoolcatchment.com.au/?cat=2`

in a feasible and accessible way; such a knowledge base requires techniques that serve as an interface for applications to infer and query about the qualitative spatial and temporal information. The field of Qualitative Spatial and Temporal Reasoning (*QSTR*) provides techniques for dealing with these issues.

QSTR has various models to encode different aspects of spatial or temporal information with qualitative relations. For example, the well-known Region Connection Calculus [97] captures the topological information between regions, such as the containment information (e.g. "Darling Harbour is *in* Sydney") and the neighbouring information (e.g. "Darling Harbour is *adjacent to* the city centre"). Directional information including "Pyrmont is *to the west* of Darling Harbour" can also be encoded with corresponding models, e.g., the Cardinal Direction Calculus [56] and the Cardinal Relation Algebra [45, 82]. Some applications including the Wikipedia "infobox" have already begun to illustrate such kind of information (see Figure 1.2 for example[4]). For temporal information, models such as the Interval Algebra [2] have been widely used for encoding relations between events, e.g. an event happens *during* another.

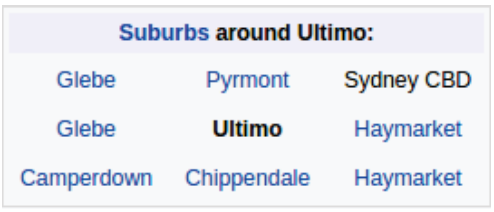


Figure 1.2: Wikipedia infobox for illustrating the directional relations between Ultimo and suburbs around it in Sydney, Australia.

For a set of spatial or temporal entities, QSTR uses qualitative constraint network (QCN) to represent the qualitative relations between them. Basically, the QCN representation consists of a set of variables for the spatial or temporal entities and a set of relations between these variables. QCNs enable a

[4]Retrieved from `https://en.wikipedia.org/wiki/Ultimo,_New_South_Wales`

knowledge base to have functions that meet various needs. For example, one fundamental function needed for many applications is to retrieve the relation between any two entities. With QCNs, a knowledge base is able to answer the questions of the users about the relations between spatial entities, such as "Does my address belong to the catchment area of X?", and "What direction is Darling Harbour to Chinatown?". By using qualitative relations in QCNs, a knowledge base can also automatically construct a qualitative description of a spatial or temporal scene such as the previous one for Darling Harbour.

Besides, with QCNs, a knowledge base can serve as an interface between qualitative information and other applications. For example, geographical databases sometimes need to adjust the geometric data of spatial objects to maintain integrity, according to some spatial constraints between these objects (see e.g. [127, 128]). Such constraints might not be explicitly available in the first place, e.g. they could be specified in documents in the form of natural language or on drawings where the relations are implicitly induced by the geometric representation. By extracting and storing the information in QCNs, the knowledge base provides geographical databases with such information in a normalized and abstract form. Another example is matching spatial scenes, where the user provides an image (e.g. the ones with a train station adjacent to a park and a lake inside the park) and wants to find other images that contain similar qualitative spatial information. By encoding the qualitative information of images in QCNs, a knowledge base can measure the similarity by comparing the qualitative relations in the QCNs (see e.g. [53]).

On the other hand, the qualitative relations in a knowledge base might have several problems. In particular, they could have conflicts, especially when the information is from human sources. The capability of detecting conflicts, or checking *consistency*, can help a knowledge base to maintain data integrity

and to detect security issues. A simple example is that the email account log-in record shows that the user is in Sydney while the phone reports to the knowledge base that the user is in London. With the additional information that Sydney and London are disjoint, the knowledge base can detect the conflict and warn the user of the potential security issue of the account. How to check consistency efficiently has been a focus of QSTR for a long time, and researchers have proposed various algorithms for this task.

Another problem of information in a knowledge base is that the spatial or temporal information at hand might be *incomplete* or *indeterminate*. For example, although the description of Darling Harbour mentions Pyrmont and Sydney central business district (CBD), we do not have explicit information about the relation between Pyrmont and Sydney CBD. QSTR provides techniques to make inferences based on partial information. The composition on relations is one of the common techniques. It can infer the possible relation between A and C if given the relation between A and B and that between B and C. In the example here, from "Pyrmont is to the west of Darling Harbour" and "Darling Harbour is to the west of Sydney CBD", it concludes that Pyrmont is also to the west of Sydney CBD.

1.2 Motivation

In the previous section, we have seen the usefulness of QSTR in building a comprehensive knowledge base for handling qualitative spatial and temporal information. However, there are important issues to resolve before we can apply QSTR to this task in practice.

In particular, a knowledge base needs to deal with large amounts of information. Therefore, the size of QCNs, i.e. the number of spatial or temporal

entities (and sometimes also the number of relations), is usually very large. For example, in order to be useful for real-world applications, a knowledge base might involve all of the points of interest in the country or even in the world, rather than just Darling Harbour. This brings about two main concerns.

The first concern is the time efficiency of reasoning techniques in QSTR to process qualitative spatial and temporal information. As mentioned before, in order to check consistency or to obtain implicit information, a knowledge base needs to perform some inferences through *qualitative reasoning* on the relations in QCNs. Researchers have devised several efficient algorithms to solve important qualitative reasoning problems, such as consistency checking and inferring strongest qualitative relations from incomplete information. These algorithms exploit specific properties of qualitative relations or structures of QCNs, or both. For these efficient algorithms to be applicable, researchers have identified several subsets of relations (e.g. convex relations discussed in the next chapter) and also useful structures of QCNs (e.g. sparse graph and chordal graph structure discussed in the next chapter). Then it is interesting to see what general properties of qualitative relations are sufficient for such algorithms to be applicable. This leads to the following question.

Question 1. Is there a more general characterisation of subclasses of qualitative calculi, for which important reasoning problems can be solved efficiently with algorithms exploiting the properties of the relations and structures of QCNs?

The second concern lies in the feasibility and the economy of the QCN representation, which affects the time and space efficiency of other applications. There are two circumstances.

The first circumstance is that the time efficiency of many applications is sensitive to the number of relations in the QCN representation. For instance,

in the previous examples of adjusting geometric data and matching spatial scenes, the efforts needed for is strongly related to the number of relations in the QCN. In such applications, the less relations are to be processed, the more efficient will these applications be. In fact, many QCNs contain unnecessary or *redundant* relations in the sense that removing such a relation from a QCN will not change the essential meaning, i.e. the solution set, of the QCN. Removing redundant relations could significantly improve the efficiency of the applications, especially when a large portion of the relations can be removed. Therefore, the following question is very important.

Question 2. How to identify redundant relations and a minimal subset of non-redundant relations that has the same solution set as the original QCN?

The second circumstance is that the QCN representation might require too much storage which makes other tasks inefficient. Here we specifically consider the task of retrieving the relation between any two spatial entities, as it is one of the fundamental functions of a knowledge base. The ideal way is to store all the relations in the QCN for the most efficient performance. However, the size of the QCN could be too large. For example, there are more than $54{,}000$ regions in the level one statistical areas of Australia and the QCN has 3 billion directional relations. The large number of relations might make the QCN too costly or even infeasible to fit into fast accessible storage (e.g. RAM), and will in turn significantly decrease the performance of retrieving relations. Therefore, we need more compact representation of qualitative relations for the task.

One compromise used by the current Geographical Information Systems (GISs) is to store the geometric representation (e.g. complex polygons) of the spatial entities and to calculate the relation directly by geometric algorithms. However, it has several drawbacks. For example, the online computation of the

spatial relations between polygons can be computationally expensive, because it requires computation time that is proportional to the number of vertices, which can be quite large (e.g. some polygons representing the country subdivisions of the USA have more than $30{,}000$ vertices). Moreover, sometimes, we may not have such geometric representation but only know how the entities are qualitatively related. The question below arises naturally.

Question 3. How to represent qualitative relations more compactly, while we can retrieve the relation between two given entities more efficiently than direct computation from geometric representation?

1.3 Contributions and Outline of the Thesis

In this thesis, we give answers to the three questions raised in the previous section. The main contributions of this thesis are summarised below. Most of the contributions have appeared in conference and journal publications.

1. For Question 1, in Chapter 3, we give a characterisation of subclasses of qualitative relations, called *distributive subalgebra*, and show that several efficient algorithms exploiting sparse structure of QCNs are applicable to solve several important qualitative reasoning problems. We also identify maximal distributive subalgebras for a number of widely used qualitative calculi, and demonstrate that some of these maximal distributive subalgebras coincide with previously identified useful subclasses in QSTR.

   Related publications:

   - Zhiguo Long and Sanjiang Li: On Distributive Subalgebras of Qualitative Spatial and Temporal, COSIT, 2015, pp. 354–374.

- Zhiguo Long, Michael Sioutis, and Sanjiang Li: Efficient Path Consistency Algorithm for Large Qualitative Constraint Networks, IJCAI, 2016, pp. 1202–1208.
- Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both: On Redundant Topological Constraints, Artificial Intelligence, 2015, vol. 225, pp. 51–76.

2. For Question 2, in Chapter 4, we propose the notion of redundant constraints in a QCN and the concept of prime subnetwork, which is a subset of a QCN without redundant constraints and has the same set of solutions as the superior QCN. Moreover, for each QCN defined over specific distributive subalgebras, we show that its prime subnetwork is unique and is exactly the set of all the non-redundant constraints in the QCN. We also devise an efficient algorithm to construct the unique prime subnetwork for such a QCN. On the other hand, we give counter examples to show that the conclusion generally does not hold for relations of other distributive subalgebras or outside a distributive subalgebra.

   Related publication:

   - Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both: On Redundant Topological Constraints, Artificial Intelligence, 2015, vol. 225, pp. 51–76.

3. For Question 3, in Chapter 5 and Chapter 6, we propose two complementary techniques to compactly represent qualitative spatial relations, and demonstrate that retrieving the relation between two entities is much more efficient than direct geometric computation. In many cases, the first approach is effective to reduce the storage of both topological and directional relations, while the second approach is a useful complement

of the first approach on cases of topological relations where the portion of non-disjoint relations is large.

Related publications:

- Zhiguo Long, Matt Duckham, Sanjiang Li, and Steven Schockaert: Indexing Large Geographic Datasets with Compact Qualitative Representation, International Journal of Geographical Information Science, 2016, vol. 30, no. 6, pp. 1072–1094.
- Zhiguo Long, Steven Schockaert, and Sanjiang Li: Encoding Large RCC8 Scenarios Using Rectangular Pseudo-Solutions, KR, 2016, pp. 463–472.

Answering these questions is an important step forward for constructing a comprehensive knowledge base, which enables applications to better handle qualitative spatial and temporal information.

To elaborate on the contributions, the remainder of the thesis is organised as follows.

- Chapter 2 introduces necessary background and basic terminology, including different models (calculi) and techniques in QSTR, and the formulation of several related problems in the language of QSTR.
- Chapter 3 discusses the properties of distributive subalgebras, and presents several efficient algorithms based on these properties. In this chapter, we will also identify the maximal distributive subalgebras of several widely used qualitative calculi in QSTR.
- Chapter 4 deals with redundancy in QCNs. We analyse the uniqueness of prime subnetworks. We devise an algorithm for constructing the unique prime subnetwork of QCNs over specific distributive subalgebras.

- Chapter 5 presents the first one of the two approaches for compactly representing qualitative spatial relations, and shows the effectiveness of this approach both in theory and in experiments on real-world datasets.
- Chapter 6 focuses on the second approach for compact representation, and demonstrates that this approach can significantly reduce the storage of topological relations and retrieve the relation efficiently by empirical evaluations on real-world datasets.
- Chapter 7 concludes the thesis and outlines promising future directions.
- Appendix A lists the maximal distributive subalgebras of PA, IA, RCC5 and RCC8.
- Appendix B gives supplementary proofs for some results in the thesis.
- Appendix C lists the publications of the author during the PhD candidature.

Chapter 2

Preliminaries and Backgrounds

2.1 Introduction

As an important subfield of Artificial Intelligence, Qualitative Spatial and Temporal Reasoning (QSTR) has been under extensive development for decades. QSTR provides various models to represent the spatial and temporal relations (e.g., adjacent to, to the north of, part of, before, and during) between entities such as indoor furniture, buildings, geographical regions, actions, and events. It also provides techniques to help extract the information from spatial or temporal scenarios. A simple example is that, given that Sydney is part of Australia and Australia is not adjacent to the UK, then with reasoning techniques in QSTR, we can immediately conclude that Sydney also is not adjacent to the UK. Moreover, from the examples in the previous chapter, we have also seen the power and the potential of QSTR in dealing with real-world problems. More discussions of the applications can be found in [24].

One major aim of QSTR is to design formal methods which could be used by non-experts to understand the physical world conveniently and efficiently. To achieve this, it needs economical representation to store the data and effi-

cient algorithms to automatically process the data, which are our two major concerns in this thesis.

In Section 2.2 and Section 2.3, we recall several well-known qualitative calculi and the concept of qualitative constraint network. Section 2.4 discusses important techniques in QSTR. Section 2.5 reviews the related work in the literature on the tasks and problems in QSTR that will be discussed in this thesis.

2.2 Qualitative Calculi

Relation is one of the most fundamental concepts in QSTR. For instance, in the examples of the previous chapter, the most frequent and essential qualitative spatial and temporal information is the qualitative relations such as "adjacent to", "to the east of", "during", and so on. Relations are used to encode the spatial or temporal information between n-*tuples* of entities such as regions and time intervals. An n-tuple is an ordered list of entities, and it is called an (ordered) *pair* when $n = 2$.

Definition 2.1 (*Relation*)**.** Let $\mathcal{D} = (D_1, \ldots, D_n)$ be a tuple of sets of entities, an n-*ary relation* R on $\mathcal{D}$ is a set of n-tuples over $\mathcal{D}$, which is a subset of $D_1 \times \ldots \times D_n$. Specifically, a *binary relation* R' on $\mathcal{D}' = (D_1, D_2)$ is a subset of (ordered) pairs over $\mathcal{D}'$.

In this thesis, we assume that the relations are all binary and that $D_1 = D_2 = \mathcal{U} \neq \varnothing$ for $\mathcal{D} = (D_1, D_2)$. The *converse* of a binary relation R is defined as $R^{-1} = \{(a_2, a_1) \in \mathcal{U} \times \mathcal{U} : (a_1, a_2) \in R\}$. The *identity relation* $\mathsf{id}_\mathcal{U}$ on $\mathcal{U}$ is defined as $\mathsf{id}_\mathcal{U} = \{(a, a) : a \in \mathcal{U}\}$. The *universal relation*, denoted by $\star$, is $\mathcal{U} \times \mathcal{U}$. When we say that a pair of entities (a_1, a_2) *satisfies* a relation R, it means $(a_1, a_2) \in R$, and for convenience we sometimes write it as $a_1 R a_2$.

The models of QSTR specify different relations between entities to encode various aspects of spatial and temporal information. The relations in a model along with some operations form a specific mathematical structure called *Boolean algebra*. A Boolean algebra is a tuple $(A, \vee, \wedge, \neg, \bot, \top)$, where A is a set, $\vee$ and $\wedge$ are two binary operations on A (called "or" and "and"), $\neg$ is a unary operation on A (called "not"), and $\bot$ and $\top$ are two distinct elements in A (called the "least" and "greatest" elements), such that the axioms of associativity, commutativity, absorption, identity, distributivity, and complements are satisfied for any elements in A. We refer the reader to [89] for more details.

Write $\mathbf{Rel}(\mathcal{U})$ for the power set of $\mathcal{U} \times \mathcal{U}$, i.e. the set of all binary relations on $\mathcal{U}$. Then $(\mathbf{Rel}(\mathcal{U}), \cup, \cap, \neg, \varnothing, \mathcal{U} \times \mathcal{U})$ is a Boolean algebra, where $\cup$, $\cap$, and $\neg$ are interpreted as the union, intersection, and complement operations of sets and $\varnothing$ is the empty set. An *atom* in this Boolean algebra is a minimal element that is not $\varnothing$, i.e. $\alpha \in \mathbf{Rel}(\mathcal{U})$ is an atom iff $\alpha \neq \varnothing$ and for every $\beta \in \mathbf{Rel}(\mathcal{U})$, either $\beta \cap \alpha = \alpha$ or $\beta \cap \alpha = \varnothing$. Suppose $\mathcal{S}$ is a subset of $\mathbf{Rel}(\mathcal{U})$, then there is a minimal subset $\widehat{\mathcal{S}}$ of $\mathbf{Rel}(\mathcal{U})$ that contains $\mathcal{S} \cup \{\varnothing, \mathcal{U} \times \mathcal{U}\}$ and is closed under the operations $\cup, \cap, \neg$. It can be seen that $(\widehat{\mathcal{S}}, \cup, \cap, \neg, \varnothing, \mathcal{U} \times \mathcal{U})$ is also a Boolean algebra, called the *Boolean subalgebra* of $(\mathbf{Rel}(\mathcal{U}), \cup, \cap, \neg, \varnothing, \mathcal{U} \times \mathcal{U})$ generated by $\mathcal{S}$.

Definition 2.2 (*Qualitative Calculus*;[83])**.** A qualitative calculus $\mathcal{M}$ on $\mathcal{U}$ is defined as a finite Boolean subalgebra of $(\mathbf{Rel}(\mathcal{U}), \cup, \cap, \neg, \varnothing, \mathcal{U} \times \mathcal{U})$. The atoms in $\mathcal{M}$ are called the *basic* relations, and we denote the set of basic relations by $\mathsf{B}_{\mathcal{M}}$.

Note that for a qualitative calculus $\mathcal{M}$, we have that $\mathsf{B}_{\mathcal{M}}$ is a set of Jointly Exhaustive and Pairwise Disjoint (*JEPD*) relations, i.e. any pair $(a, b) \in \mathcal{U} \times \mathcal{U}$ satisfies exactly one of the relations in $\mathsf{B}_{\mathcal{M}}$, and the union of all the relations in $\mathsf{B}_{\mathcal{M}}$ is the universal relation $\star$. Given a set of JEPD relations $\mathsf{B}_{\mathcal{M}}$, then there

is a Boolean algebra that is generated by $\mathsf{B}_{\mathcal{M}}$. This Boolean subalgebra is a qualitative calculus. Therefore, a set of JEPD relations defines a qualitative calculus and vice versa.

A (qualitative) relation R in a qualitative calculus is a union of basic relations in $\mathsf{B}_{\mathcal{M}}$, i.e. $R = \alpha_{i_1} \cup \ldots \cup \alpha_{i_k}$ for some $\alpha_{i_1}, \ldots, \alpha_{i_k} \in \mathsf{B}_{\mathcal{M}}$. Note that for convenience we will sometimes write R as the set $\{\alpha_{i_1}, \ldots, \alpha_{i_k}\}$, while it still means $R = \alpha_{i_1} \cup \ldots \cup \alpha_{i_k}$. The notation $\alpha \in R$ will be used specifically for the case $\alpha \in \mathsf{B}_{\mathcal{M}}$ and $\alpha \subseteq R$. A subset of the relations $\mathcal{S}$ in a qualitative calculus $\mathcal{M}$ is called a *subclass* of $\mathcal{M}$.

In the following, we will introduce several of the most widely used qualitative calculi by specifying the set of basic relations $\mathsf{B}_{\mathcal{M}}$ for them. For a more comprehensive survey of qualitative calculi, we refer the readers to [33].

Definition 2.3 (Point Algebra [125])**.** Let $\mathcal{U}$ be the set of real numbers. The Point Algebra (*PA*) is the Boolean subalgebra generated by the JEPD set of relations $\{<, >, =\}$, where $<, >$ and $=$ are the natural orderings of real numbers.

PA contains eight relations, viz. the three basic relations $<, >$, and $=$, the empty relation, the universal relation $\star$, and three non-basic relations $\{<, =\}$, $\{>, =\}$, and $\{<, >\}$ (i.e. $\leq, \geq, \neq$). It can be used to characterise temporal relations between actions. For example, in the definition of "offside offence" in association football (i.e. soccer) from Wikipedia[1], it says that if the action A ("the player is in an offside position and is actively involved in the play") and the action B ("the ball is touched by a team mate") happen at the same moment, then it is an offside offence. The condition in the definition can be modelled by PA as $A = B$.

[1] https://en.wikipedia.org/wiki/Offside_(association_football)#Offside_offence

Table 2.1: IA basic relations between intervals x and y.

Relation	Symbol	Converse	Definition
before	b	bi	$x^+ < y^-$
meets	m	mi	$x^+ = y^-$
overlaps	o	oi	$x^- < y^- < x^+ < y^+$
starts	s	si	$x^- = y^- < x^+ < y^+$
during	d	di	$y^- < x^- < x^+ < y^+$
finishes	f	fi	$y^- < x^- < x^+ = y^+$
equals	eq	eq	$x^- = y^- < x^+ = y^+$

Definition 2.4 (Interval Algebra [2])**.** Let $\mathcal{U}$ be the set of closed intervals on the real line. Thirteen binary relations between two intervals $x = [x^-, x^+]$ and $y = [y^-, y^+]$ are defined by the order of the four endpoints of x and y (see Table 2.1). The Interval Algebra (*IA*) is generated by these JEPD relations.

IA is much more expressive than PA. It contains $2^{13} = 8192$ relations in total, which can be used to express the relations between time intervals (e.g. events), such as the relation "while" or *during* (d) between the events A ("the victim was crying") and B ("the suspect was smashing down the door").

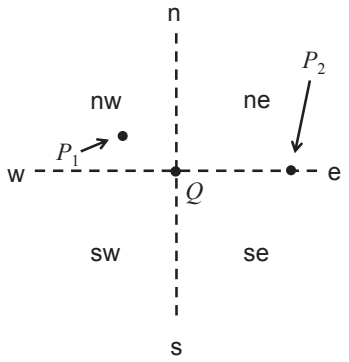
Definition 2.5 (Cardinal Relation Algebra [45, 82])**.** Let $\mathcal{U}$ be the real plane. Define relations nw, n, ne, w, eq, e, sw, s and se between two points x and y as in Fig. 2.1a. The Cardinal Relation Algebra (*CRA*) is generated by these nine JEPD relations.

CRA can be viewed as an extension of PA to the points in the plane and is able to express the directional relations between points, such as "the bus station is to the north-east (ne) of the hotel". Each of the basic relations of CRA is actually a pair of the PA relations between the x-coordinates and between the y-coordinates. In other words, any basic CRA relation r is the *Cartesian product* of some basic relations of PA, i.e. $r = \{(a, b) : a = (x, y) \wedge b = (x', y') \wedge (x, x') \in \alpha \wedge (y, y') \in \beta\}$. For convenience, we use $\alpha \otimes \beta$ to denote the Cartesian product

of two basic relations α, β in PA or IA, i.e. $\alpha \otimes \beta = \{(a,b) : a = (x,y) \wedge b = (x',y') \wedge (x,x') \in \alpha \wedge (y,y') \in \beta\}$, where each of x, y, x', y' can be a point or an interval on the x/y-axis..

Relation	Definition
nw	$x < x', y > y'$
n	$x = x', y > y'$
ne	$x > x', y > y'$
w	$x < x', y = y'$
eq	$x = x', y = y'$
e	$x > x', y = y'$
sw	$x < x', y < y'$
s	$x = x', y < y'$
se	$x > x', y < y'$

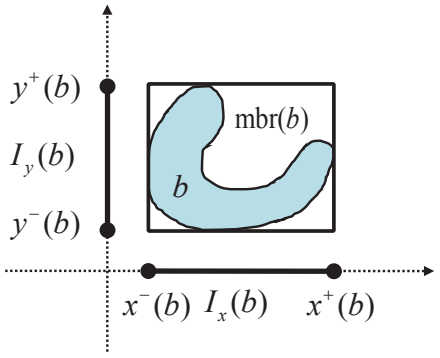
(a) Basic relations of CRA.



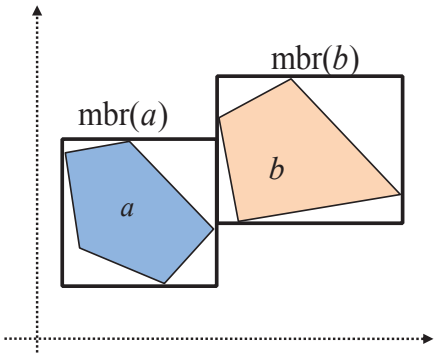
(b) Examples: P_1 nw Q and P_2 e Q.

Figure 2.1: Definition of basic CRA relations and example of CRA relations.

More generally, for two arbitrary relations R and S in PA or IA, we will write $R \otimes S$ for the relation $\bigcup\{\alpha \otimes \beta : \alpha \in R, \beta \in S, \alpha, \beta \in \mathsf{B}\}$. Note that the converse of a relation $R \otimes S$ is exactly the same as $R^{-1} \otimes S^{-1}$. Analogously, for two *subclasses* of relations $\mathcal{R}$ and $\mathcal{S}$ in $\mathcal{M}$, we will write $\mathcal{R} \otimes \mathcal{S}$ for the set of relations $\{R \otimes S : R \in \mathcal{R}, S \in \mathcal{S}\}$.



(a) The MBR mbr(b) of a region b.



(b) The RA relation of the MBRs is m $\otimes$ o.

Figure 2.2: Illustration of MBR and RA relation.

Similarly, IA can be extended to deal with relations between the *regions* in the plane. A *region* a in the real plane is a bounded non-empty *regular closed* set of points, where a point set is regular closed if it is identical to the closure of its interior, i.e. $a = \overline{(a^\circ)}$, where $\cdot^\circ$ is the interior operator and $\bar{\cdot}$ is the closure operator in the Euclidean plane topological space. Let us assume that an orthogonal basis is given in the plane. For a region a, let x_a^- (or y_a^-) denote $\inf\{x : (\exists y)(x,y) \in a\}$ (or $\inf\{y : (\exists x)(x,y) \in a\}$) and x_a^+ (or y_a^+) denote $\sup\{x : (\exists y)(x,y) \in a\}$ (or $\sup\{y : (\exists x)(x,y) \in a\}$). Then the *minimum bounding rectangle* (*MBR*) of a, denoted by $\mathsf{mbr}(a)$, is the point set $\{(x,y) : x_a^- \leq x \leq x_a^+$ and $y_a^- \leq y \leq y_a^+\}$. In other words, $\mathsf{mbr}(a)$ is the smallest *axis-aligned* rectangle that contains a. A rectangle is axis-aligned if its sides are parallel to the axes of a given Cartesian coordinate system in the plane. Figure 2.2 illustrates this. We write $I_x(a) = [x_a^-, x_a^+]$ and $I_y(a) = [y_a^-, y_a^+]$ as the x- and y-projections of $\mathsf{mbr}(a)$ respectively. Then $\mathsf{mbr}(a)$ can also be represented as $(I_x(a), I_y(a))$.

Definition 2.6 (Rectangle Algebra [6])**.** Given two regions a and b, the basic rectangle relation between them is $\alpha \otimes \beta$ iff $(I_x(a), I_x(b)) \in \alpha$ and $(I_y(a), I_y(b)) \in \beta$, where $\alpha, \beta \in \mathsf{B}_{\mathrm{IA}}$. We write B_{RA} for the set of basic rectangle relations, i.e.,

$$\mathsf{B}_{\mathrm{RA}} = \{\alpha \otimes \beta : \alpha, \beta \in \mathsf{B}_{\mathrm{IA}}\}. \tag{2.1}$$

The Rectangle Algebra (*RA*) is defined on the set of the MBRs of the regions in the plane and is generated by the relations in B_{RA}.

There are 169 different basic RA relations in B_{RA}. As it uses the MBRs of regions to define the relation, RA can be used to approximate the relations between regions such as the following topological relations and directional relations. By extending RA into domains of higher dimensions, Balbiani et al. [7]

also proposed the Block Algebra (*BA*) to characterize the relations between hyper-rectangles.

The Region Connection Calculus (*RCC*) was first proposed by Randell et al. in [97] as a first order theory, and there is an equivalent definition which uses the Boolean connection algebra [118]. The latter uses the connection relation **C** to define the other relations. The same set of relations has also been identified for regions in geographic information science [38, 39, 117].

In this thesis, we are only concerned with the standard interpretation for RCC, that is, the universe $\mathcal{U}$ is the set of regions in the plane, and the connection relation **C** is defined as $\mathbf{C} = \{(a, b) \in \mathcal{U} \times \mathcal{U} : a \cap b \neq \varnothing\}$. The other relations can be defined in a similar way accordingly. Table 2.2 gives the meaning of these relations under this interpretation. See [84] for more details.

Table 2.2: Topological interpretation of basic RCC8 relations in the plane, where a, b are two regions, and a° and b° are the interiors of a and b, respectively.

Relation	Meaning	Relation	Meaning
DC	$a \cap b = \varnothing$	**TPP**	$a \subset b$, $a \not\subset b^\circ$
EC	$a \cap b \neq \varnothing$, $a^\circ \cap b^\circ = \varnothing$	**NTPP**	$a \subset b^\circ$
PO	$a \not\subseteq b$, $b \not\subseteq a$, $a^\circ \cap b^\circ \neq \varnothing$	**EQ**	$a = b$

Definition 2.7 (RCC5 and RCC8)**.** Let $\mathcal{U}$ be the set of regions in the real plane. The calculus RCC8 is generated by the eight relations of topological information

$$\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{EQ}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}.$$

RCC5 is the sub-algebra of RCC8 generated by the five part-whole relations

$$\mathbf{DR}, \mathbf{PO}, \mathbf{EQ}, \mathbf{PP}, \mathbf{PP}^{-1},$$

where $\mathbf{DR} = \mathbf{DC} \cup \mathbf{EC}$, $\mathbf{PP} = \mathbf{TPP} \cup \mathbf{NTPP}$, and $\mathbf{PP}^{-1} = \mathbf{TPP}^{-1} \cup \mathbf{NTPP}^{-1}$. See

Figure 2.3 for the illustration of these relations.

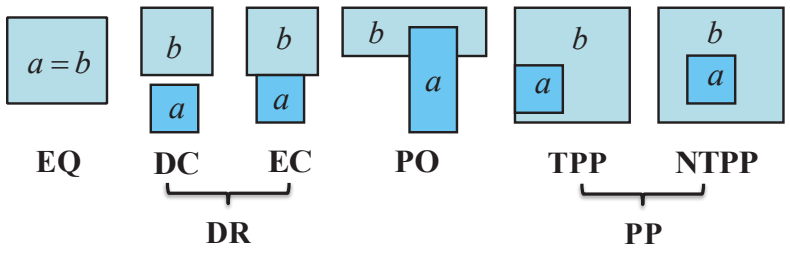


Figure 2.3: Illustration for basic relations in RCC5 / RCC8

We should note that all of the above qualitative calculi PA, IA, CRA, RA, and RCC5/8 contain the identity relation $\mathsf{id}_{\mathcal{U}}$ and are closed under converse.

The following calculus for directional relations was first proposed by Goyal and Egenhofer [56] in the context of geographical information science (GIS) for *connected regions*. A region is connected if it is not the union of two regions which are disjoint with each other. Later, the extension of this calculus was proposed by Skiadopoulos and Koubarakis in [116] for possibly disconnected regions.

Definition 2.8 (Cardinal Directional Calculus)**.** The Cardinal Directional Calculus (*CDC*) makes use of the MBR of b to encode the directional information of a target region a to a reference region b. By extending the four edges of $\mathsf{mbr}(b)$, it decomposes the plane into nine tiles, named as NW, N, NE, W, O, E, SW, S, and SE (see Figure 2.4), and represents the relation $\delta(a, b)$ from a to b as a subset of $T = \{\mathsf{NW}, \mathsf{N}, \mathsf{NE}, \mathsf{W}, \mathsf{O}, \mathsf{E}, \mathsf{SW}, \mathsf{S}, \mathsf{SE}\}$, where a tile name, say NW, is in $\delta(a, b)$ if and only if an interior point of a is in tile NW (see Figure 2.4 for illustration). The set of basic relations in CDC consists of all of the possible relations between (connected or arbitrary) regions that is a subset of T.

We call $\delta(a, b)$ a *k-tile relation* if it contains k tile names, and specifically we call a 1-tile relation a *single tile relation*. For convenience, we also use the tile

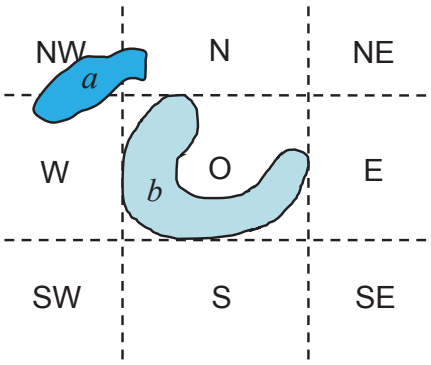


Figure 2.4: The 9 tiles of b and illustration of CDC relation $\delta(a, b) = \{\mathsf{W}, \mathsf{NW}, \mathsf{N}\}$.

name in the single tile relation $\delta(a, b)$ to represent this CDC relation, e.g., NW for the relation $\{\mathsf{NW}\}$.

Note that unlike the other calculi before, CDC is not closed under converse. If only connected regions are considered, then there are 218 basic relations in CDC; if arbitrary bounded regions are considered, then there are 511 basic relations in CDC. See [86, 131] and [56] for more information about CDC.

Up to now, we have introduced the qualitative calculi used in this thesis for representing the spatial or temporal information. These calculi are among the most popular ones in the context of literature and applications. There are many other calculi that are also useful. For example, for directional information, the orientation of the object can be important, and the calculi $\mathcal{LR}$ [80, 108] and $\mathcal{OPRA}_m$ [91] are suitable for representing this kind of information. The discussion of these calculi, however, will not be covered in this thesis. We refer the interested readers to, e.g., [24].

Next, we will have a look at how QSTR formally captures the information of a spatial or temporal scene and how to extract implicit information by reasoning techniques. Here, a spatial or temporal scene could be a set of regions in a map or some text descriptions of the relations between places or events.

2.3 Qualitative Constraint Network

The previously introduced qualitative calculi define qualitative relations for encoding the information between two spatial or temporal entities. For two variables representing spatial or temporal entities, a *constraint* specifies a relation between them. A *variable* represents spatial or temporal entities.

Definition 2.9 (Constraint)**.** A *constraint* C is a pair (V, R), where $V = (u, v)$ is a pair of variables and R is a relation defined on the domain of the variables. In this case, V is called the *scope* of the constraint C and we say that C (or the corresponding relation R) relates u to v. For ease of representation, we use (uRv) to denote the constraint $((u, v), R)$.

For a spatial or temporal scene, usually there are several variables and constraints. Qualitative constraint networks are used to capture the information in a structured way.

Definition 2.10 (Qualitative Constraint Network)**.** Given a subclass $\mathcal{S}$ of relations in a qualitative calculus $\mathcal{M}$, a *qualitative constraint network* (QCN) over $\mathcal{S}$ is a tuple $(V, \mathcal{C})$, where $V = \{v_1, \ldots, v_n\}$ is a non-empty finite set of spatial or temporal variables and $\mathcal{C}$ is a finite set of constraints in the form $(v_i R_{ij} v_j)$ with $R_{ij} \in \mathcal{S}$ and $v_i, v_j \in V$.

In this thesis, for a QCN $\mathcal{N}$ over a subclass $\mathcal{S}$ of $\mathcal{M}$, we require that for each pair (v_i, v_j) of variables

- there exists at most one constraint in $\mathcal{N}$ that relates v_i to v_j;
- if the relation for (v_i, v_j) with $v_i \neq v_j$ is unspecified, i.e. there is no constraint in $\mathcal{N}$ that relates v_i to v_j, then the universal relation $\star$ in $\mathcal{M}$ relates v_i to v_j;

- when v_i is the same as v_j, the constraint in $\mathcal{N}$ that relates v_i to itself is the identity relation $\mathsf{id}_{\mathcal{U}}$ for PA, IA, CRA, RA, RCC5 or RCC8 (e.g., **EQ** in RCC5/8), or O for CDC.

Also, unless specified otherwise, we assume that $\mathcal{N}$ is not *trivially inconsistent*, i.e. $\forall v_i \neq v_j, (v_i \varnothing v_j) \notin \mathcal{N}$.[2] For a QCN $\mathcal{N} = (V, \mathcal{C})$, if V is clear in the context, the notation $\mathcal{N}$ and the notation $\mathcal{C}$ for the set of constraints will be interchangeably used. Note that for some calculi, the relation in $\mathcal{N}$ that relates v_i to v_j might not be the converse of the relation that relates v_j to v_i. This is the case for CDC, as CDC is not closed under converse [86, 131]. For a QCN $\mathcal{N}$ over PA, IA, CRA, RA, or RCC5/8 with variables V, we always assume that $R_{ij} = R_{ji}^{-1}$ for any $v_i, v_j \in V$, and when specifying a QCN $\mathcal{N}$ we only include one of the constraints C_{ij} and C_{ji}.

Example 2.1. $\mathcal{N} = (V, \mathcal{C})$, where $V = \{v_1, v_2, v_3, v_4\}$ and $\mathcal{C} = \{(v_1\{\textbf{DC}, \textbf{PO}\}v_2), (v_2\{\textbf{EC}\}v_3), (v_3\{\textbf{NTPP}\}v_4), (v_1\{\textbf{TPP}^{-1}\}v_4)\}$, is a QCN over RCC8 relations.

A QCN $\mathcal{N} = (V, \mathcal{C})$ is called a *complete network* if $\forall v_i, v_j \in V$ $(v_i \neq v_j)$ the constraint relating v_i to v_j is specified in $\mathcal{C}$. Furthermore, $\mathcal{N}$ is called a *scenario* if it is a complete network and $\forall v_i, v_j \in V$ $(v_i \neq v_j)$ and $(v_i R_{ij} v_j) \in \mathcal{C}$ we have $R_{ij} \in \mathsf{B}_{\mathcal{M}}$. For $\mathcal{N} = (V, \mathcal{C})$, a certain scenario $\mathcal{N}' = (V, \mathcal{C}')$, which satisfies that $\forall (v_i R_{ij} v_j) \in \mathcal{C}$ we have $\delta_{ij} \in R_{ij}$ where $(v_i \delta_{ij} v_j) \in \mathcal{C}'$, is called a scenario of $\mathcal{N}$. For instance, the QCN in the above example is not a complete network, and is not a scenario, because the constraints $(v_1 R_{13} v_3)$ and $(v_2 R_{24} v_4)$ are not in $\mathcal{C}$ and the relation {**DC**,**PO**} relating v_1 and v_2 is not basic. In this thesis, we use the term *basic network* to refer to a possibly incomplete network in which the relation of every constraint is either a basic relation or the universal relation $\star$.

Next, we introduce some derivatives of a QCN.

[2] In literature, this is also known as 2-consistency. See [47] for more details.

Definition 2.11 (Restriction)**.** Given a QCN $\mathcal{N}$ over a qualitative calculus $\mathcal{M}$ for variables $V = \{v_1, v_2, ..., v_n\}$, the *restriction of* $\mathcal{N}$ *on* $V_0 \subseteq V$ is $\mathcal{N}|_{V_0} = \{(uRv) \in \mathcal{N} : u, v \in V_0\}$.

A *partial scenario* of a QCN $\mathcal{N} = (V, \mathcal{C})$ for variables $V_0 \subseteq V$ is a scenario of the restriction of $\mathcal{N}$ on V_0. For the QCN in Example 2.1, $\mathcal{N}|_{V_0} = \{(v_2\{\textbf{EC}\}v_3), (v_3\{\textbf{NTPP}\}v_4), (v_2\{\textbf{TPP}^{-1}\}v_4)\}$, where $V_0 = \{v_2, v_3\}$, is a partial scenario of $\mathcal{N}$. Note that the relation between v_2 and v_4 becomes $\textbf{TPP}^{-1}$, while previously it is not specified in $\mathcal{N}$ (i.e. it is $\star$).

Sometimes, we would like to *refine* the original QCN for extracting useful information.

Definition 2.12 (Refinement)**.** Let $\mathcal{N} = (V, \mathcal{C})$ and $\mathcal{N}' = (V, \mathcal{C}')$ be QCNs. We say $\mathcal{N}'$ *refines* $\mathcal{N}$ if $\forall(v_i R_{ij} v_j) \in \mathcal{C}$ we have $R'_{ij} \subseteq R_{ij}$ where $(v_i R'_{ij} v_j) \in \mathcal{C}'$. In this case, $\mathcal{N}'$ is called a *refinement* of $\mathcal{N}$.

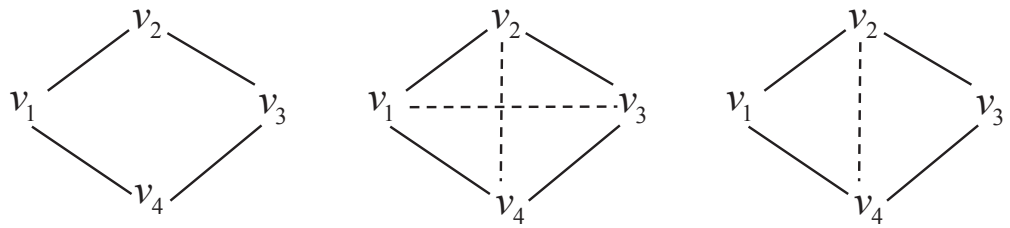
The QCN $\mathcal{N}' = \{(v_1\{\textbf{DC}\}v_2), (v_2\{\textbf{EC}\}v_3), (v_3\{\textbf{NTPP}\}v_4), (v_2\{\textbf{EC}, \textbf{TPP}^{-1}\}v_4), (v_1\{\textbf{TPP}^{-1}\}v_4)\}$ is a refinement of $\mathcal{N}$ in Example 2.1. Here, the relation between v_1 and v_2 has been refined to **DC** and that between v_2 and v_4 becomes $\{\textbf{EC}, \textbf{TPP}^{-1}\}$.

The word "network" in the name of QCN already indicates that there is a graph structure associated with a QCN.

Definition 2.13 (Constraint Graph)**.** The (underlying) *constraint graph* of a QCN $\mathcal{N} = (V, \mathcal{C})$, denoted by $G_{\mathcal{N}} = (V, E)$, is a graph that has the variables of $\mathcal{N}$ as its set of vertices, and a set of edges $E = \{\{v_i, v_j\} : (v_i R_{ij} v_j) \in \mathcal{C}$ where $R_{ij} \neq \star$ and $v_i \neq v_j\}$.

For the QCN $\mathcal{N}$ in Example 2.1, the constraint graph of $\mathcal{N}$ is the one shown in Figure 2.5(a). The complete graph shown in Figure 2.5(b) is the completion

of the constraint graph of $\mathcal{N}$, where the relations between the vertices of added edges are the universal relation $\star$.



(a) Constraint Graph (b) Complete Graph (c) Chordal Graph

Figure 2.5: Illustrations of the graph structures related to a QCN.

Instead of completing a constraint graph, we can transform it into a *triangulated* or *chordal graph*. The concept of *cycle* is important for a chordal graph. A cycle in an undirected graph is a sequence of vertices $(v_{i_1}, v_{i_2}, \ldots, v_{i_k})$ where $v_{i_1} = v_{i_k}$ and $v_{i_r} \neq v_{i_s}$ for any $1 \leq r \neq s \leq k$ ($\{r, s\} \neq \{1, k\}$), and $\{v_{i_r}, v_{i_{r+1}}\}$ ($1 \leq r \leq k$, $i_{k+1} = i_1$) is an edge in the graph. The length of a cycle is the number of unique vertices, e.g., $k - 1$ for the previously mentioned cycle. A *chord* in a cycle is an edge connecting two non-consecutive vertices of the cycle. For example, in the graph in Figure 2.5(b), $(v_1, v_2, v_3, v_4, v_1)$ is a cycle of length 4 and $\{v_2, v_4\}$ is a chord.

Definition 2.14 (Chordal Graph; [12])**.** An undirected graph $G = (V, E)$ is *chordal* if every cycle of length greater than three has a chord.

The graph in Figure 2.5(c) is a chordal graph that is transformed from the constraint graph in Figure 2.5(a).

Given a graph $G = (V, E)$, for each $v \in V$, the adjacency set $\mathsf{adj}(v)$ is defined as $\{w \in V : \{v, w\} \in E\}$. For the graph in Figure 2.5(c), $\mathsf{adj}(v_2) = \{v_1, v_3, v_4\}$. A vertex v is *simplicial* if $\mathsf{adj}(v)$ induces a complete graph. In Figure 2.5(c), the vertices v_2 and v_4 are not simplicial, while v_1 and v_4 are simplicial. Every chordal graph has a simplicial vertex (cf. [12]). Moreover, after removing

a simplicial vertex and its incident edges from a chordal graph, the resulting subgraph remains chordal. The order in which simplicial vertices of sequential subgraphs are successively removed is called a *perfect elimination ordering*. Formally, suppose that $(v_1, \ldots, v_{n-1}, v_n)$ is an ordering to remove the vertices in G, and denote by F_k the set $\{v_j \in \text{adj}(v_k) : j > k\}$. The ordering is a perfect elimination ordering if F_k induces a complete subgraph of G for every k. For example, (v_1, v_2, v_3, v_4) is a perfect elimination ordering for the chordal graph in Figure 2.5(c). This is because, v_1 is simplicial in the initial chordal graph and $F_1 = \{v_2, v_4\}$ induces a complete subgraph; v_2 is simplicial in the resulting chordal graph by removing v_1 and the associated edges in the initial chordal graph, and $F_2 = \{v_3, v_4\}$ is a complete subgraph of G; the cases for v_3 and v_4 are similar. Each chordal graph has a perfect elimination ordering and a graph having a perfect elimination ordering is a chordal graph (cf. [12]). Note that a complete graph of any order is also a chordal graph, e.g. the one in Figure 2.5(b). A non-chordal graph can be made chordal or *triangulated* by, following a variable elimination ordering, adding the missing edges in F_k to the graph s.t. F_k becomes a complete subgraph of G for every k. For a graph $G = (V, E)$, the *maximum cardinality search* algorithm [119] can be used to find a variable elimination ordering for triangulating G by adding f edges in time $O(|V| + |E| + f)$, and if the graph is already chordal then the algorithm will return a perfect elimination ordering without adding any new edges. Take the constraint graph in Figure 2.5(a) as a running example. The algorithm first arbitrarily selects a vertex, say v_1. Then it selects a vertex in the rest with the largest number of neighbours in the set of previously selected vertices (i.e. $\{v_1\}$). v_2 and v_4 are such vertices and here we select v_2. For v_3 and v_4, note that v_3 has one neighbour (v_1) in the set of previously selected vertices, and v_4 also has one neighbour (v_2). Thus we can choose v_3, and then v_4, and obtain

an elimination ordering (v_1, v_2, v_3, v_4). The triangulated graph corresponding to this ordering is the one in Figure 2.5(c).

2.4 Important Techniques in QSTR

2.4.1 Weak Composition

After introducing the basic models and the QCN representation in QSTR, now we are able to discuss in more detail about the techniques and problems in QSTR. They are useful for answering the questions raised in the previous chapter.

The *composition* operation for two relations is a well-known basic inference rule. It is usually used to infer the relation between A and C when the relation between A and B and that between B and C are given. For two relations R and S, the composition $R \circ S$ is defined as $\{(x, y) : \exists z \text{ s.t. } (x, z) \in R \wedge (z, y) \in S\}$. In QSTR, for each calculus, there is a so-called *composition table* which specifies the results of the "compositions" of relations in the calculus. However, researchers (e.g., [99]) have long been aware that the composition table for a qualitative calculus sometimes does not correspond to the usual definition of composition. In fact, the usual composition of two relations in a qualitative calculus $\mathcal{M}$ might not be a relation in $\mathcal{M}$ any more. The "composition" used in several qualitative calculi is actually a weaker notion, that is, the *weak composition*. For example, Li and Ying [78] have shown that the "composition" is always weak composition under any interpretation of RCC8, e.g., although **PO** is contained in the weak composition of **EC** and **TPP**, **PO** is not contained in the composition of **EC** and **TPP**. A simple example [78] for this is as follows. Given that a, b and c are three pairwise disjoint rectangles, let $x = a \cup b$ and $y = b \cup c$ ($(x, y) \in$ **PO**). Then there does not exist a region z s.t. $(x, z) \in$ **EC**

and $(z, y) \in$ **TPP**.

Table 2.3: Composition table for PA relations.

$\diamond$	$<$	$>$	$=$
$<$	$<$	$<,>,=$	$<$
$>$	$<,>,=$	$>$	$>$
$=$	$<$	$>$	$=$

Formally, the *weak composition* of two basic relations α and β in $\mathsf{B}_{\mathcal{M}}$ of a qualitative calculus, denoted by $\alpha \diamond \beta$, is defined as $\bigcup\{\gamma \in \mathsf{B}_{\mathcal{M}} : \gamma \cap (\alpha \circ \beta) \neq \varnothing\}$. When the relations (say R and S) are not basic, the weak composition of them is defined as $R \diamond S = \bigcup\{\alpha \diamond \beta : \alpha \in R, \beta \in S\}$. For PA, IA, CRA, and RA, the weak composition operation in these calculi is exactly the same as the composition operation (see e.g. [99]). Note that for CRA and RA, the (weak) composition of two basic relations $(\alpha \otimes \beta) \circ (\gamma \otimes \delta)$ is the same as $(\alpha \circ \gamma) \otimes (\gamma \circ \delta)$, where $\alpha, \beta, \gamma, \delta$ are basic relations of PA or IA. The composition table of PA, RCC5, and RCC8 are given in Tables 2.3, 2.4, and 2.5. The items (e.g. "$<, >, =$") in the composition tables represent the union of the relations in the corresponding item. For IA, we refer to [2]. For CRA and RA, each of their composition tables can be obtained from the composition tables of PA and IA respectively. For CDC, Liu et al. [86] has designed an algorithm to generate its composition table.

Table 2.4: Composition table for RCC5 relations.

$\diamond$	**DR**	**PO**	**PP**	**PP**$^{-1}$	**EQ**
DR	**DR,PO,PP,PP**$^{-1}$**,EQ**	**DR,PO,PP**	**DR,PO,PP**	**DR**	**DR**
PO	**DR,PO,PP**$^{-1}$	**DR,PO,PP,PP**$^{-1}$**,EQ**	**PO,PP**	**DR,PO,PP**$^{-1}$	**PO**
PP	**DR**	**DR,PO,PP**	**PP**	**DR,PO,PP,PP**$^{-1}$**,EQ**	**PP**
PP^{-1}	**DR,PO,PP**$^{-1}$	**PO,PP**$^{-1}$	**PO,PP,PP**$^{-1}$**,EQ**	**PP**$^{-1}$	**PP**$^{-1}$
EQ	**DR**	**PO**	**PP**	**PP**$^{-1}$	**EQ**

With weak composition, converse, and intersection of relations, we can define *subalgebras* of a qualitative calculus $\mathcal{M}$. A *subalgebra* $\mathcal{S}$ of $\mathcal{M}$ contains all

Table 2.5: Composition table for RCC8 relations.

$\diamond$	DC	EC	PO	TPP	NTPP	TPP^{-1}	NTPP^{-1}	EQ
DC	DC,EC,PO TPP,NTPP TPP^{-1},EQ NTPP^{-1}	DC,EC PO TPP NTPP	DC,EC PO TPP NTPP	DC,EC PO TPP NTPP	DC,EC PO TPP NTPP	DC	DC	DC
EC	DC,EC,PO TPP^{-1} NTPP^{-1}	DC,EC,PO EQ,TPP TPP^{-1}	DC,EC,PO TPP NTPP	EC,PO TPP NTPP	PO TPP NTPP	DC EC	DC	EC
PO	DC,EC,PO TPP^{-1} NTPP^{-1}	DC,EC,PO TPP^{-1} NTPP^{-1}	DC,EC,PO TPP,TPP^{-1},EQ NTPP,NTPP^{-1}	PO TPP NTPP	PO TPP NTPP	DC,EC,PO TPP^{-1} NTPP^{-1}	DC,EC,PO TPP^{-1} NTPP^{-1}	PO
TPP	DC	DC EC	DC,EC PO,TPP NTPP	TPP NTPP	NTPP	DC,EC,PO EQ,TPP TPP^{-1}	DC,EC,PO TPP^{-1} NTPP^{-1}	TPP
NTPP	DC	DC	DC,EC PO TPP NTPP	NTPP	NTPP	DC,EC PO TPP NTPP	DC,EC,PO TPP,TPP^{-1} NTPP,EQ NTPP^{-1}	NTPP
TPP^{-1}	DC,EC,PO TPP^{-1} NTPP^{-1}	EC,PO TPP^{-1} NTPP^{-1}	PO TPP^{-1} NTPP^{-1}	PO,EQ TPP TPP^{-1}	PO TPP NTPP	TPP^{-1} NTPP^{-1}	NTPP^{-1}	TPP^{-1}
NTPP^{-1}	DC,EC,PO TPP^{-1} NTPP^{-1}	PO TPP^{-1} NTPP^{-1}	PO TPP^{-1} NTPP^{-1}	PO TPP^{-1} NTPP^{-1}	PO,TPP,EQ NTPP,TPP^{-1} NTPP^{-1}	NTPP^{-1}	NTPP^{-1}	NTPP^{-1}
EQ	DC	EC	PO	TPP	NTPP	TPP^{-1}	NTPP^{-1}	EQ

the basic relations and a subset of non-basic relations in $\mathcal{M}$, and is *closed* under weak composition, converse, and intersection. Note that a set of relations is closed under converse (weak composition or intersection) if the converse of any relation (the weak composition or the intersection of any two relations) in the set is still in it. The minimal subalgebra under set inclusion that contains $\text{B}_{\mathcal{M}}$ is called the *closure* of the basic relations. The ORD-Horn subclass $\mathcal{H}$ of IA is a subalgebra of IA, $\hat{\mathcal{H}_5}$ and $\hat{\mathcal{H}_8}$ are subalgebras of RCC5 and RCC8 respectively, and $\mathcal{H} \otimes \mathcal{H}$ and the strongly-preconvex subclass [6] are two subalgebras of RA and the latter strictly contains the former.

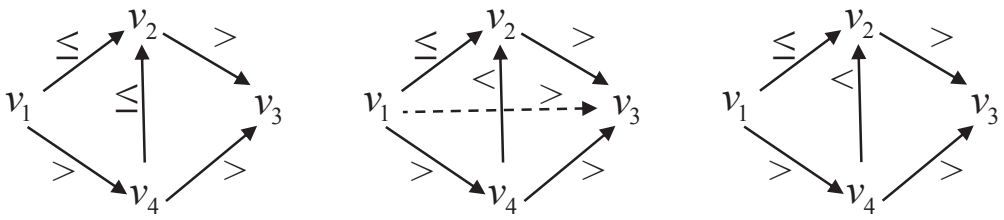
2.4.2 Path Consistency and Partial Path Consistency

The weak composition operation provides a basic inference rule, which however is not enough for more sophisticated inferencing tasks. Therefore, re-

searchers proposed more advanced techniques for such tasks, such as *path consistency*.

Definition 2.15 (Path Consistency)**.** A QCN $\mathcal{N} = (V, \mathcal{C})$ is *path consistent* (**PC**) iff $\forall v_i, v_k, v_j \in V$, we have that $\varnothing \neq R_{ij} \subseteq R_{ik} \diamond R_{kj}$, where R_{ij}, R_{ik}, and R_{kj} are the relations that relates (v_i, v_j), (v_i, v_k), and (v_k, v_j) in $\mathcal{N}$.

For example, Figure 2.6(a) shows a PA QCN $\mathcal{N}$ which is not path consistent, while Figure 2.6(b) shows a path consistent QCN.



(a) $\mathcal{N}$ (b) $\mathcal{N}_{\text{p}}$ (c) $\mathcal{N}_{\text{p}}^{G}$

Figure 2.6: A PA QCN $\mathcal{N}$, its a-closure $\mathcal{N}_{\text{p}}$, and its partially path consistent network $\mathcal{N}_{\text{p}}^{G}$.

Note that path consistency was originally defined by Montanari [90] for "paths" in the constraint graph of a constraint network, rather than for *triangles* (e.g. $\{v_i, v_k, v_j\}$) as defined here. For CSPs (and similarly for qualitative calculi), Montanari [90] has shown that the definition for "triangles" is equivalent to the original definition by Montanari. For QCNs, similar conclusion holds, as Renz and Ligozat [99] have proved that using weak composition instead of composition in the definition of path consistency does not change the essence for the calculi discussed here.

There were several algorithms in the literature to achieve **PC** (e.g. [90, 121, 122]). In this thesis, we adopt the one used in [121] which runs in $O(n^3)$ time to decide if a QCN is path consistent, where n is the number of variables in the QCN. The algorithm (i.e. PC) is shown in Algorithm 1. It is worth

noting that, in the algorithm, we only consider triples of distinct variables (Lines 1 and 11). Nevertheless, the algorithm will still achieve $\mathbf{PC}$, under the assumption that any input QCN $\mathcal{N} = (\{v_1, v_2\}, \mathcal{C})$ with only two variables is not trivially inconsistent, i.e. $(v_1 \varnothing v_2), (v_2 \varnothing v_1) \notin \mathcal{C}$. In Section 2.3, we have assumed that any of the considered QCNs is not trivially inconsistent unless specified otherwise. Therefore in the rest of the thesis, we will regard PC as a fully functioning algorithm for achieving $\mathbf{PC}$. If the algorithm did not return inconsistency after enforcing PC on a QCN $\mathcal{N}$, then the resulting $\mathbf{PC}$ QCN $\mathcal{N}_{\mathrm{p}}$ is called the *a-closure* of $\mathcal{N}$. For example, in Figure 2.6, $\mathcal{N}_{\mathrm{p}}$ is the a-closure of $\mathcal{N}$.

Algorithm 1: PC($\mathcal{N}$), a path consistency algorithm.

Input: $\mathcal{N}$, a QCN with constraints $\mathcal{C} = \{(v_i R_{ij} v_j) : 1 \leq i, j \leq n\}$.
Output: $\mathcal{N}_{\mathrm{p}}$, the a-closure of $\mathcal{N}$, or inconsistency.

1 $Q \leftarrow \bigcup_{1 \leq i < j \leq n} \{(i, j, k), (k, i, j) : 1 \leq k \leq n, k \neq i, k \neq j\}$;
2 **while** $Q \neq \varnothing$ **do**
3 select and delete a path (i, k, j) from Q;
4 tempR $\leftarrow R_{ij} \cap (R_{ik} \diamond R_{kj})$;
5 **if** tempR $= \varnothing$ **then**
6 **return** inconsistency;
7 **end**
8 **if** tempR $\subset R_{ij}$ **then**
9 $R_{ij} \leftarrow$ tempR;
10 $R_{ji} \leftarrow \text{tempR}^{-1}$;
11 $Q \leftarrow Q \cup \{(i, j, k), (k, i, j) : 1 \leq k \leq n, k \neq i, k \neq j\}$;
12 **end**
13 **end**
14 $\mathcal{N}_{\mathrm{p}} \leftarrow \mathcal{N}$;
15 **return** $\mathcal{N}_p$.

The fundamental idea of PC is to call the following rule until the QCN becomes stable or a relation becomes the empty relation.

$$R_{ij} \leftarrow (R_{ik} \diamond R_{kj}) \cap R_{ij}. \tag{2.2}$$

For dynamic information, there are also incremental versions of PC. In [48], Gerevini devised such algorithms for PA and IA, while for RCC8 Sioutis and Condotta [110] proposed a vertex incremental one. PC has significantly improved our ability to solve the reasoning problems in QSTR. However, sometimes enforcing PC is an overkill, especially for QCNs that are in a sparse structure.

Bliek and Sam-Haroud [13] first proposed the so-called *partial path consistency* to solve CSPs by exploiting the sparse structure. Chmeiss and Condotta [18], as well as Sioutis and Koubarakis [112], adopted this notion for QSTR.

Definition 2.16 (Partial Path Consistency)**.** Let $\mathcal{N} = (V, \mathcal{C})$ be a QCN and $G = (V, E)$ be a chordal graph such that $G_{\mathcal{N}} \subseteq G$, where $G_{\mathcal{N}}$ is the constraint graph of $\mathcal{N}$. $\mathcal{N}$ is *partially path consistent* (**PPC**) w.r.t. G iff $\forall\{v_i, v_j\}$, $\{v_i, v_k\}$, $\{v_k, v_j\} \in E$, we have that $\varnothing \neq R_{ij} \subseteq R_{ik} \diamond R_{kj}$, where R_{ij}, R_{ik}, and R_{kj} are the relations that relates (v_i, v_j), (v_i, v_k), and (v_k, v_j) in $\mathcal{N}$.

As an example, the QCN $\mathcal{N}_{\mathrm{p}}^{G}$ in Figure 2.6(c) is **PPC** w.r.t. the chordal graph in the figure, and it is not **PC**.

Similar to the case of **PC**, **PPC** was originally defined for "paths" in an arbitrary graph G by Bliek and Sam-Haroud [13] rather than for the "triangles" in a chordal graph. Nevertheless, as shown in [13] when the G is a chordal graph, this definition will be equivalent to the original one. For a QCN $\mathcal{N}$, the algorithm PPC in Algorithm 2 decides **PPC** on $\mathcal{N}$, and the resulting QCN $\mathcal{N}_{\mathrm{p}}^{G}$ is called the *partially path consistent subnetwork* of $\mathcal{N}$ w.r.t. G. For example, the QCN $\mathcal{N}_{\mathrm{p}}^{G}$ in Figure 2.6(c) is the partially path consistent subnetwork of $\mathcal{N}$ w.r.t. the graph shown in the figure. Note that when the chordal graph G is a complete graph on V, establishing **PPC** is exactly the same as establishing **PC**, and the two algorithms PPC and PC are identical in this case. In the next

chapter, we will present a more efficient algorithm to establish PPC as well as PC.

In the following, we will see that establishing **PC** or **PPC** on a QCN can solve several major problems in QSTR.

Algorithm 2: PPC($\mathcal{N}, G$), a partial path consistency algorithm.

Input: $\mathcal{N}$, a QCN with constraints $\mathcal{C} = \{(v_i R_{ij} v_j) : 1 \leq i, j \leq n\}$; G, a chordal graph s.t. $G_{\mathcal{N}} \subseteq G$.
Output: $\mathcal{N}_{\mathrm{p}}^{G}$, the partially path consistent subnetwork of $\mathcal{N}$, or inconsistency.

```
1  Q ← {{i,j,k} : {v_i,v_j},{v_i,v_k},{v_j,v_k} ∈ E(G)};
2  while Q ≠ ∅ do
3      select and delete a triangle {i,k,j} from Q;
4      for each permutation of (v_i,v_k,v_j) do
5          tempR ← R_ij ∩ (R_ik ⋄ R_kj);
6          if tempR = ∅ then
7              return inconsistency;
8          end
9          if tempR ⊂ R_ij then
10             R_ij ← tempR;
11             R_ji ← tempR^-1;
12             Q ← Q ∪ {{i,k,j} : {v_i,v_k},{v_j,v_k} ∈ E(G_N)};
13         end
14     end
15 end
16 N_p^G ← N;
17 return N_p^G.
```

2.5 Tasks and Problems in QSTR

In the following, we will take a look at some sophisticated tasks in QSTR that are important for an application to handle qualitative spatial or temporal information.

2.5.1 The Consistency Problem

The first important reasoning task is to check the *consistency* of a QCN. This task originates from Constraint Satisfaction Problems (CSPs). A QCN can be considered as a special instance of CSP and the consistency problem naturally arises for qualitative calculi. Moreover, as we have seen in the introduction of this thesis, checking consistency is an important function in applications as well, e.g. to check if there are conflicts in the data or to detect abnormal activities.

Before formally introducing the task of consistency checking, we first define what is a solution of a QCN.

Definition 2.17 (Solution, Consistency, Equivalence)**.** Given a QCN $\mathcal{N}$ over a qualitative calculus $\mathcal{M}$ on $\mathcal{U}$ for variables $V = \{v_1, v_2, ..., v_n\}$ and an assignment $\sigma : V \to \mathcal{U}$, we say that σ is a *solution* of $\mathcal{N}$ if for any $1 \leq i, j \leq n$ $(\sigma(v_i), \sigma(v_j)) \in R_{ij}$, where $(v_i R_{ij} v_j)$ is a constraint in $\mathcal{N}$ that relates v_i to v_j. For simplicity, we usually refer to a solution σ as the sequence of values $(\sigma(v_1), \ldots, \sigma(v_n))$. We say $\mathcal{N}$ is *consistent* or *satisfiable* if it has a solution. A QCN $\mathcal{N}_1$ is *equivalent* to another QCN $\mathcal{N}_2$ over the same set of variables if they have the same set of solutions.

Intuitively, a solution is an illustration of the qualitative relations at hand. For example, when the QCN is about the topological relations between school catchment areas and neighbourhoods, then the regions in the map for these areas form a solution of the QCN.

Rather than assigning values to all the variables in a QCN, sometimes a partial assignment for a subset of variables is also useful.

Definition 2.18 (Partial Solution)**.** Given a QCN $\mathcal{N}$ over a qualitative calculus $\mathcal{M}$ on $\mathcal{U}$ for variables $V = \{v_1, v_2, ..., v_n\}$, a *partial solution* on $V_0 \subseteq V$ of $\mathcal{N}$ is

a solution of the QCN that is the restriction of $\mathcal{N}$ on V_0, i.e. $\mathcal{N}|_{V_0} = \{(uRv) \in \mathcal{N} : u, v \in V_0\}$.

Formally, the task of checking the consistency of QCNs, also known as *the consistency problem*, can be defined as follows.

Problem 1 (The Consistency Problem)**.** Let $\mathcal{S}$ be a subclass of a qualitative calculus $\mathcal{M}$. The consistency problem for $\mathcal{S}$ is defined as:

> **Instance:** A QCN $\mathcal{N} = (V, \mathcal{C})$, where V is a finite set of variables, and $\mathcal{C}$ is a finite set of constraints $(v_i R_{ij} v_j)$ that relate variables in V s.t. $R_{ij} \in \mathcal{S}$.
>
> **Question:** Decide the consistency of $\mathcal{N}$.

The consistency problem for many calculi turns out to be quite hard in general. For example, for each of IA, CRA, RA, RCC5/8 and CDC, the consistency problem for the whole calculus is NP-complete [6, 82, 86, 98, 100, 125]. For these calculi, there are different branches of research that aim to identify appropriate conditions such that the consistency problem becomes *tractable*, i.e. polynomial algorithms exist to solve it. One of the branches is to restrict the structure of the QCN, such as requiring it to be of bounded treewidth [15, 64] or to be in a chordal graph structure [18, 112]. Another branch is to limit the allowed relations in the QCN, i.e. the subclass $\mathcal{S}$. These two branches can be combined together to identify tractable problems. There is a similar phenomenon in the research of CSPs (cf. [22]).

In particular, for restricting the structure of the QCN, in [14], Bodirsky and Dalmau showed that QCNs in IA that have bounded treewidth can be solved in polynomial time. Later in [15], Bodirsky and Wölfl demonstrated that the conclusion also applies to the RCC8 calculus. Huang et al. [64] proposed another sufficient condition (i.e. aNAP) for the tractability of the consistency problem

of bounded treewidth QCNs and showed that several important calculi (including IA, RA and RCC8) satisfy this condition. The research in [18, 112] on the consistency problem of chordal graph structured QCNs has proposed several efficient backtracking algorithms to decide the consistency of such QCNs for calculi including IA and RCC8. Sioutis et al. [114] also proposed an efficient algorithm for the consistency problem of chordal structured QCNs over distributive subalgebras.

For many qualitative calculi, people have identified several large tractable subclasses, where polynomial algorithms exist to solve the consistency problem. Some of these subclasses are *maximal*, in the sense that adding any other relation in the calculus to the subclass will result in it being intractable. In particular, for IA, there are exactly eighteen maximal tractable subclasses [71], while only one of them (i.e. the ORD-Horn subclass, denoted as $\mathcal{H}$) contains all the basic relations of IA, as shown in [93]; for RCC8, Renz and Nebel [98, 100] have identified all the three maximal tractable subclasses containing all the basic relations (including a horn subclass denoted by $\hat{\mathcal{H}}_8$); for RCC5, Jonsson and Drakengren [65] provided a complete classification of the approximately four billion subclasses into tractable and NP-complete ones, and identified all the four maximal tractable subclasses, among which only the horn subclass $\hat{\mathcal{H}}_5$ contains all the basic relations; for CDC, Liu et al. [86] showed that it is polynomial for the consistency problem of complete basic networks, while it remains an open question what would be the maximal tractable subclasses for CDC; for CRA, Ligozat [82] has shown that the preconvex subclass is a maximal tractable subclass; the subclass $\mathcal{H} \otimes \mathcal{H}$ of RA is tractable following from the tractability of $\mathcal{H}$, and Balbiani et al. [6] found a maximal tractable subclass of RA, called the strongly-preconvex subclass. Similar results also apply to the generalized rectangle algebra, i.e. the Block Algebra (BA; [7]). On the other

hand, the whole calculus of PA is tractable [72, 122], and the researchers also identified a *pointisable* subclasses of IA (denoted as SA) contained in $\mathcal{H}$, in which each relation can be transformed into a conjunction of several PA relations. The above tractable subclasses will be repeatedly mentioned later, and we assume that the term *tractable subclasses* henceforth specifically refers to these ones.

For these tractable subclasses, people proposed various efficient algorithms to solve the consistency problem. For example, for PA and SA (i.e. pointisable IA), van Beek [121] developed an $O(n^2)$ time algorithm ($n = |V|$ is the number of variables in a QCN), by making use of topological sort [69]. For CDC, Liu et al. [86] gave an $O(n^3)$ time algorithm ($n = |V|$) that constructs a canonical solution that only uses integers for complete basic CDC QCNs. These algorithms only work for specific calculi. More often, local consistencies play an important role in solving the consistency problem, especially $\mathbf{PC}$ and $\mathbf{PPC}$ defined previously.

On one hand, $\mathbf{PC}$ is a necessary condition for consistency. The algorithm PC returns inconsistency only if the QCN is not consistent. On the other hand, $\mathbf{PC}$ is not always able to solve the consistency problem of general calculi, e.g., the whole calculi IA, RCC5/8, and RA. Nevertheless, for some subclasses of the calculi, it is a sufficient condition. In fact, establishing $\mathbf{PC}$ is sufficient to decide the consistency problem of the tractable subclass $\mathcal{H}$ for IA [93], the three maximal tractable subclasses (including $\hat{\mathcal{H}}_8$) for RCC8 and $\hat{\mathcal{H}}_5$ for RCC5 [100], the preconvex subclass of CRA [82], the strongly-preconvex subclass of RA [6], and the whole calculus PA [72, 121]. The algorithm PC has also been applied for solving the consistency problem of the whole calculi, as the forward checking technique in a backtracking algorithm (cf. [92]).

As $\mathbf{PC}$, $\mathbf{PPC}$ is also a necessary but not always sufficient condition for

consistency, and the algorithm PPC returns inconsistency only if the QCN is not consistent. Interestingly, there are also subclasses that $\mathbf{PPC}$ is enough to solve the consistency problem. For IA and RCC5/8, Chmeiss and Condotta [18], as well as Sioutis and Koubarakis [111, 112], showed that establishing $\mathbf{PPC}$ is sufficient and more efficient than establishing $\mathbf{PC}$ to decide the consistency of QCNs over the subclasses $\mathcal{H}$, $\hat{\mathcal{H}}_5$ and $\hat{\mathcal{H}}_8$. Also, these authors have used PPC to help solve the consistency algorithm of the whole calculi more efficiently with a backtracking algorithm. Sioutis and Condotta [109, 110] devised an incremental version of PPC for IA and RCC8 to solve the consistency problem for dynamic information. Later in Chapter 3, we will see that there is more efficient algorithm than PC and PPC to solve the consistency problem of QCNs over distributive subalgebras.

2.5.2 The Minimal Labelling Problem

When extracting the relation between two entities from a spatial or temporal scene configuration, we might be interested in the best knowledge that can be inferred from the given information. For example, given that "the city centre of Sydney is not *equal* to the Sydney central business district", we might ask whether the latter is contained in the former, especially when we have other qualitative spatial information about these two regions.

In terms of QSTR, given a QCN, we would like to find out the *feasible* basic relations between any two variables.

Definition 2.19 (Feasibility and Minimality)**.** Let $\mathcal{N} = (V, \mathcal{C})$ be a QCN. For a constraint $(v_i R_{ij} v_j)$ in $\mathcal{N}$ and a basic relation α in R_{ij}, we say α is *feasible* if there exists a solution $(a_1, a_2, \ldots, a_n)$ of $\mathcal{N}$ such that $(a_i, a_j) \in \alpha$. We say that $\mathcal{N}$ is *minimal* if every basic relation α in R_{ij} is feasible for every pair of variables (v_i, v_j), and in this case R_{ij} is called the *minimal label* of (v_i, v_j).

Generally, for a QCN $\mathcal{N}$, some of the relations might not be minimal labels. Removing all the infeasible basic relations results in a refinement of $\mathcal{N}$, called the *minimal subnetwork* of $\mathcal{N}$, denoted by $\mathcal{N}_{\mathrm{m}}$. Formally, a minimal QCN that refines $\mathcal{N}$ and is equivalent to $\mathcal{N}$ is the minimal subnetwork of $\mathcal{N}$. Figure 2.7(a) shows a PA QCN $\mathcal{N}$ which is not minimal, because the basic relation $=$ between v_1 and v_3 is not feasible (i.e. no solution of $\mathcal{N}$ satisfies it). The QCN $\mathcal{N}_{\mathrm{m}}$ in Figure 2.7(b) is the minimal subnetwork of $\mathcal{N}$.

(a) $\mathcal{N}$ (b) $\mathcal{N}_{\mathrm{m}}$

Figure 2.7: A PA QCN $\mathcal{N}$ and its minimal subnetwork $\mathcal{N}_{\mathrm{m}}$.

One of the fundamental tasks in QSTR is to find the minimal subnetwork of a given QCN.

Problem 2 (The Minimal Labelling Problem)**.** Let $\mathcal{S}$ be a subalgebra of a qualitative calculus $\mathcal{M}$. The minimal labelling problem for $\mathcal{S}$ is defined as:

> **Instance:** A QCN $\mathcal{N} = (V, \mathcal{C})$, where V is a finite set of variables, and $\mathcal{C}$ is a finite set of constraints $(v_i R_{ij} v_j)$ relating variables in V s.t. $R_{ij} \in \mathcal{S}$.
>
> **Task:** Compute the minimal subnetwork of $\mathcal{N}$.

For general calculus, this problem is NP-hard [85]. For tractable subclasses, the minimal labelling problem is also tractable, because we can check the feasibility of each basic relation in each of the constraints in polynomial time and there are only finite many constraints in a QCN. However, such method to compute the minimal subnetwork is not efficient, i.e. $O(n^5)$ for previously discussed tractable subclasses. Note that in general, the previous algorithms PC

and PPC cannot transform a QCN over those tractable subclasses to a minimal one. There are some examples for that in [100, 121, 122].

Researchers tried to develop more efficient algorithms than the naive one of time complexity $O(n^5)$. van Beek [121] devised an $O(n^3)$ time algorithm to compute the minimal network of a general QCN over PA, and Gerevini and Schubert [50] fixed a faulty in the proof of van Beek. Gerevini also proposed efficient algorithms to incrementally solve the minimal labelling problem of PA and IA in less than $O(n^5)$ time in most cases. Later, for sparsely structured QCN over the convex subclass of PA relations that do not contain the relation $\{\neq\}$, Gerevini and Saetti [49] devised a more efficient algorithm that makes use of time graphs and meta-graphs and runs in $O(n^2)$ time. Amaneddine et al. [4, 5] proposed backtracking algorithms for general IA and RCC8 QCNs, with the help of smaller tractable subclasses which have the so-called patchwork property [63] and the property that $\mathbf{PPC}$ implies the consistency of a QCN.

Some research also focused on identifying subclasses of relations where a QCN that is $\mathbf{PC}$ is also minimal. In the context of CSPs, Montanari [90] gave a specific characterization of relations (i.e. monotone relations) that fall in these subclasses, which however do not cover all the qualitative relations discussed here. Valdés-Pérez [120] proved that complete and basic QCNs over IA have the property that $\mathbf{PC}$ implies minimality. van Beek [122] showed that QCNs over convex PA relations also have this property. Ligozat [81] identified the subclass $\mathcal{C}_{\text{IA}}$ of convex relations in IA that possesses such property. Later, Chandra and Pujari [17] identified a similar subclass of RCC8 which also has the property. Recently, Amaneddine and Condotta [3] identified two subclasses of IA with such property. Later in this thesis, we will present our work on distributive subalgebras which unifies such subclasses of all the calculi discussed here (except CDC).

2.5.3 The (Weakly) Global Consistency Problem

In addition to having a QCN representing the relations between entities, people sometimes would like to have a solution or a consistent scenario for the QCN. For example, in real-world applications, after specifying the relations between objects, people might be interested in seeing a sketch map (i.e. a solution) illustrating these relations [66]. Also, when the given QCN contains incomplete or indeterminate information, people might want to have a refinement of the QCN that only contains complete and determinate information (i.e. a consistent scenario). However, to construct a solution of a consistent QCN or even of a minimal QCN might still be very hard. In fact, Liu and Li [85] showed that for CRA, IA and RCC8, it is NP-hard to compute even a single solution of a minimal QCN over these calculi. This is also the case for constructing a consistent scenario of a minimal QCN, because for these calculi, once a consistent scenario is given, constructing a solution takes polynomial time [82, 85, 121]. On the other hand, with *(weakly) global consistency*, we can construct a solution (or a consistent scenario) for a QCN in polynomial time in an incremental and backtrack-free manner. As a result, (weakly) global consistency becomes quite important for making the task easier for specific QCNs.

Definition 2.20 (Global Consistency)**.** We say a QCN $\mathcal{N} = (V, \mathcal{C})$ is *globally consistent* if any partial solution of $\mathcal{N}$ on any non-empty subset V' of V can be extended to a solution of $\mathcal{N}$.

For example, the QCN $\mathcal{N}_{\mathrm{m}}$ in Figure 2.7(b) is globally consistent. This is because, with any partial solution of it, say (a_1, a_2) for v_1 and v_2 where $a_1 < a_2$, we can always extend it to a solution, e.g., (a_1, a_2, a_3) with $a_3 > a_1$ and $a_3 = a_2$.

Note that the "composition" for RCC5/8 is weak composition (cf. the example in Section 2.4.1), then it immediately follows that some partial solution

of a QCN over RCC5/8 cannot be extended to a solution of the whole QCN. Therefore, to deal with this circumstance, we proposed the concept of *weakly global consistency*.

Definition 2.21 (Weakly Global Consistency)**.** We say a QCN $\mathcal{N} = (V, \mathcal{C})$ is *weakly globally consistent* if any partial consistent scenario of $\mathcal{N}$ on any $V' \subseteq V$ can be extended to a consistent scenario of $\mathcal{N}$.

Note that weakly global consistency is about extending a partial scenario, while global consistency is about extending a partial solution. Nevertheless, it is easy to see that they are equivalent for qualitative calculi where weak composition is exactly composition. For example, the globally consistent QCN $\mathcal{N}_{\mathrm{m}}$ in Figure 2.7 is also weakly globally consistent. This is because we can extend any partial scenario of $\mathcal{N}_{\mathrm{m}}$, say $\{(v_1 < v_2)\}$, to a consistent scenario of $\mathcal{N}_{\mathrm{m}}$, e.g., $\{(v_1 < v_2), (v_1 < v_3), (v_2 < v_3)\}$.

The *(weakly) globally consistent subnetwork* of $\mathcal{N}$ is the refinement of $\mathcal{N}$ that is equivalent to $\mathcal{N}$ and is (weakly) globally consistent. The corresponding problem of computing the (weakly) globally consistent subnetwork of a QCN is formulated as below.

Problem 3 (The (Weakly) Global Consistency Problem)**.** Let $\mathcal{S}$ be a subalgebra of a qualitative calculus $\mathcal{M}$. The (weakly) global consistency problem is defined as:

> **Instance:** A QCN $\mathcal{N} = (V, \mathcal{C})$, where V is a finite set of variables, and $\mathcal{C}$ is a finite set of constraints $(v_i R_{ij} v_j)$ relating variables in V s.t. $R_{ij} \in \mathcal{S}$.
>
> **Task:** Compute the (weakly) globally consistent subnetwork of $\mathcal{N}$.

Note that by definition (weakly) global consistency implies minimality. It has been noted by researchers that sometimes $\mathbf{PC}$ will achieve globally consistency of a QCN. Bessière et al. [11] and Ligozat [81] showed that $\mathcal{C}_{\mathrm{IA}}$, the

previously mentioned subclass of convex IA relations, indeed has the stronger property that QCNs over $\mathcal{C}_{\text{IA}}$ that are $\mathbf{PC}$ are also globally consistent. That is, establishing $\mathbf{PC}$ on such QCNs solves the global consistency problem for them. For each of the calculi IA and PA, Amaneddine and Condotta [3] found that there are exactly two maximal subclasses that have this property, i.e. establishing $\mathbf{PC}$ on QCNs over such subclass decides the global consistency. There are also some special subclasses in the context of CSP (e.g., row-convex relations [123] and tree-convex relations [132]) possess this property. For the convex subclass of RCC8 relations identified in [17], establishing $\mathbf{PC}$ on the QCNs over that subclass cannot ensure the global consistency. However, we will present in the next chapter that establishing $\mathbf{PC}$ on these QCNs ensures the weakly global consistency. More generally, we will show that $\mathbf{PC}$ can solve the (weakly) global consistency problem for QCNs over distributive subalgebras of the calculi discussed in this thesis.

One may have noticed that the previous literature had a strong emphasis on the time efficiency aspect of QSTR, while paying little attention to the problem of representing the information economically. The efficient reasoning techniques developed so far are all based on the QCN representation which encodes the relations between variables or entities with constraints. However, the QCN representations have at least two problems: 1. A QCN might contain constraints that are unnecessary or *redundant*; 2. The representation can cost a large storage space, as a complete QCN contains $\Theta(n^2)$ number of constraints. In the following, we discuss these two problems of QCN in more detail.

2.5.4 The Redundancy Problem

For the first problem of the QCN representation, we introduce the concept of *redundancy* w.r.t. *entailment*. Let $\mathcal{N} = (V, \mathcal{C})$ be a QCN. We say $\mathcal{N}$ *entails*

a constraint (uRv), denoted by $\mathcal{N} \models (uRv)$, if every solution σ of $\mathcal{N}$ satisfies (uRv), i.e. $(\sigma(u), \sigma(v)) \in R$. A constraint (uRv) in $\mathcal{N}$ is *redundant* if the network $(V, \mathcal{C} \setminus \{(uRv)\})$ entails (uRv). $\mathcal{N}$ is *prime* if it does not have redundant constraints. Suppose $\mathcal{N}' = (V, \mathcal{C}')$ is a QCN where $\mathcal{C}' \subseteq \mathcal{C}$. If $\mathcal{N}'$ is prime and equivalent to $\mathcal{N}$, then we say $\mathcal{N}'$ is a *prime subnetwork* of $\mathcal{N}$.

By the definition, each universal constraint $(v_i \star v_j)$ in $\mathcal{N}$ is redundant in $\mathcal{N}$, and hence we call it a *trivial* redundant constraint. The following shows an example of non-trivial redundant constraint.

Example 2.2. Suppose

$$\mathcal{N} = \{v_1 \,\mathbf{PP}\, v_2, v_1 \,\mathbf{PP}\, v_5, v_3 \,\mathbf{PP}\, v_1, v_4 \,\mathbf{PP}\, v_2, v_5\{\mathbf{DR}, \mathbf{PP}\}v_2, v_3\mathbf{PO}v_4\},$$

as illustrated in Figure 2.8.

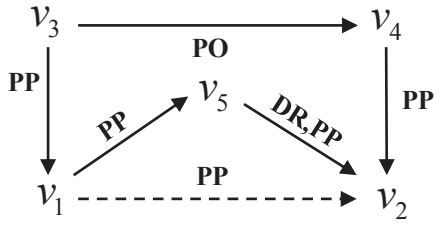


Figure 2.8: An RCC5 QCN where $(v_1 \,\mathbf{PP}\, v_2)$ is redundant.

Then $(v_1 \,\mathbf{PP}\, v_2)$ is redundant. It is because, after establishing PC to $\mathcal{N} \setminus \{(v_1 \,\mathbf{PP}\, v_2)\}$, we have $(v_5 \,\mathbf{PP}\, v_2)$ and hence $(v_1 \,\mathbf{PP}\, v_2)$. This shows that $\mathcal{N} \setminus \{(v_1 \,\mathbf{PP}\, v_2)\}$ entails $(v_1 \,\mathbf{PP}\, v_2)$. Moreover, $(v_1 \,\mathbf{PP}\, v_2)$ is the only non-trivial redundant constraint in $\mathcal{N}$ and $\mathcal{N} \setminus \{(v_1 \,\mathbf{PP}\, v_2)\}$ is the unique prime subnetwork of $\mathcal{N}$.

Removing redundant constraints in QCNs can result in a sparser QCN and accelerate other tasks. For example, in geospatial databases, some geometrical objects were misplaced or in wrong shape and we need to adjust them so

that the spatial constraints between objects are satisfied. Wallgrün [127] gave a solution to this problem by transforming the problem into an optimisation problem with constraints encoded as numeric inequations. The cost of solving the optimisation problem is strongly correlated to the number of explicit topological constraints. By using prime subnetworks, redundant constraints are removed so that the number of inequations is reduced to a relatively small number. In fact, Wallgrün used two approximate algorithms to remove redundant constraints, which however are not guaranteed to result in a network with no redundant constraints.

Prime subnetworks might also be able to reduce the complexity of the comparison between different spatial or temporal scenes. In applications such as image search, the spatial relations between components in the image serve as the semantic representation of a image. To match similar images or patterns, one heuristic is based on the spatial relations between the components (see, e.g., [53, 94]). For example, a matching query of images could be to find images in which a park contains a lake and some boats, and the boats are on the lake. The entities are "park" (P), "lake" (L) and "boats" (B). Encoded by the RCC5 model, the spatial constraints between them are L**PP**P, B**PP**P, and B**PP**L. The matching process would be to find the images that have a similar set of constraints as the specified ones. The similarity of two set of constraints (i.e. QCNs) $\mathcal{N}$ and $\mathcal{N}'$ is measured by computing the distance between the two QCNs. A natural definition of the distance is given as follows (see, e.g., [25, 75, 129]):

$$\mathsf{dist}(\mathcal{N},\mathcal{N}') = \sum\{\mathsf{dist}(R,R') : (xRy) \in \mathcal{N} \text{ and } (xR'y) \in \mathcal{N}'\},$$

where $\mathsf{dist}(R,R')$ is defined as the number of changes needed to transform R to R' in the conceptual neighbourhood graph (see [46] and Section 3.7.1). The

number of comparisons of relations would be $\Theta(n^2)$ if $\mathcal{N}$ and $\mathcal{N}'$ are complete QCNs. However, note that there might be some redundant constraints in $\mathcal{N}$ and $\mathcal{N}'$, that is, the constraints that can be inferred from the other constraints in the QCN. For instance, in the previous example, the constraint $B\mathbf{PP}P$ is actually redundant because we can infer this constraint by using $L\mathbf{PP}P$ and $B\mathbf{PP}L$. If we first remove these redundant constraints, and obtain two prime subnetworks of $\mathcal{N}$ and $\mathcal{N}'$ (denoted by $\mathcal{N}_{\mathrm{pr}}$ and $\mathcal{N}'_{\mathrm{pr}}$), then the distance becomes

$$\begin{aligned}&\mathsf{dist}_{\mathrm{pr}}(\mathcal{N},\mathcal{N}')\\&=\sum\{\mathsf{dist}(R,R'):(xRy)\in\mathcal{N},(xRy)\in\mathcal{N}',((xRy)\in\mathcal{N}_{\mathrm{pr}}\text{ or }(xR'y)\in\mathcal{N}'_{\mathrm{pr}})\}.\end{aligned}$$

If $\mathcal{N}$ and $\mathcal{N}'$ contain many redundant constraints, then the new distance will significantly reduce the number of comparisons.

By reducing the density of the constraint graph, the prime subnetwork technique could also serve as a preprocessing technique for QCNs to gain trade-off in future computation time. One possible application is for reasoning with dynamic information. Suppose that we have a QCN which is rather dense. We need to update the QCN with new information frequently and to perform qualitative reasoning on the updated QCN. Propagating the information in the original QCN requires reasoning on a dense graph, which could be slow. By contrast, after removing redundant constraints in the original QCN, the resulting prime subnetwork could be sparsely structured, and will need less efforts, as there are many algorithms (e.g. PPC) for reasoning more efficiently on sparser QCNs.

To obtain a prime subnetwork, the fundamental problem is the redundancy problem, formulated as follows.

Problem 4 (The Redundancy Problem)**.** Let $\mathcal{S}$ be a subalgebra of a qualitative

calculus $\mathcal{M}$. The redundancy problem is defined as:

> **Instance:** A QCN $\mathcal{N} = (V, \mathcal{C})$, where V is a finite set of variables, and $\mathcal{C}$ is a finite set of constraints $(v_i R_{ij} v_j)$ relating variables in V s.t. $R_{ij} \in \mathcal{S}$; a constraint $(uRv) \in \mathcal{C}$.
>
> **Task:** Decide if (uRv) is redundant in $\mathcal{N}$.

The redundancy problem has already been noticed in other research, especially the ones outside QSTR. Years ago, Ginsberg [52], and later Schmolze and Snyder [106], proposed algorithms to deal with redundancy in knowledge bases. Gottlob and Fermuller [55] considered removing redundancy from logical clause and Liberatore [79] analysed the problem for CNF formulae. Grimm and Wissmann [58] addressed redundancy checking of ontologies. Chmeiss et al. [19] studied redundancy in CSPs w.r.t. a given local consistency. Egenhofer and Sharma [40] first considered the redundancy problem for QSTR, and gave a lower bound and upper bound of the size of a prime subnetwork, while they did not provide any efficient algorithm for constructing a prime subnetwork. Rodríguez et al. [105] considered this problem for query processing with respect to complete basic RCC8 QCNs. Recently, Wallgrün [127] devised two approximate algorithms to identify and remove redundant constraints in a QCN, which however cannot guarantee to simplify the QCN optimally. In fact, as shown in Chapter 4, to efficiently identify a prime subnetwork with the minimum number of constraints is also a challenge. Similar to the case of CNF formulae [79], some QCNs might have many prime subnetworks. Nevertheless, later in Chapter 4 we will show that for RCC5/8 and PA, the prime subnetwork of a QCN over a distributive subalgebra would be unique and can be found in $O(n^3)$ time. Sioutis et al. [113] have extended the results to sparsely structured QCNs.

2.5.5 The Compact Representation Problem

The task of retrieving the relation between two given entities is a fundamental problem for both QSTR and geographical information systems (GISs), as well as many real-world applications. There is a growing need for integrating qualitative spatial (and temporal) information into databases [43, 57, 105] and modern technologies like digital maps and recommendation systems [16, 37, 73], such that the information can be efficiently retrieved.

In QSTR, reasoning techniques rely on the availability of the relation between any variables. In GISs, query answering is a central function. Queries about qualitative spatial information include checking the relation between objects and finding instances of regions that satisfy a given spatial constraint (see [43] for a summary of different types of queries). We focus on the type of queries about checking the relation between regions, which can be regarded as the most fundamental one among the other types. For example, the type of queries about finding all regions that satisfy a relation with a given region, the essence is to obtain the actual relation between any two regions, which is the query type we focus on.

The QCN representation could be too large to be suitable for retrieving the relation between any two entities. For example, as mentioned in Chapter 1, there are more than $54,000$ regions in the statistical areas of Australia (Level 1) and the QCN for CDC information would have 3 billion relations. The large number of relations might make the QCN too costly or even infeasible to fit into fast accessible storage (e.g. RAM), and will in turn significantly decrease the performance of the QCN for the task of retrieving the relation.

The comprise in GISs is to use the geometric representation (e.g. complex polygons that might have holes and/or multiple connected components). This geometric representation scales linearly with the number of regions while the

relation is calculated at query time. Clementini et al. [20, 21] studied using the 9-intersection model developed by Egenhofer et al. [38, 39] to determine the topological relation between two regions, which have been adopted by the current GISs. The computation of spatial relation between two regions can be very expensive. In fact, in many cases we need to check if two polygons intersect, which requires a computation time that is proportional to the number of vertices, e.g. $O(m \log m)$ for simple polygons [104] where m is the number of vertices and m can be quite large (e.g. some polygons representing the country subdivisions of the USA have more than $30{,}000$ vertices). Also, for some regions, we may only know how they are qualitatively related to other regions, without having access to precise boundaries. This is often the case with vernacular places [124, 126], as well as in applications that rely on extracting spatial information from natural language [107]. For example, the school catchment areas sometimes are specified by text descriptions of the relations between the areas and the neighbourhoods in the form like "the catchment area of School A contains the neighbourhoods to the west of the train station".

Considering the above, the following problem arises.

Problem 5 (The Compact Representation Problem)**.**

> **Instance:** A set of regions or a set of qualitative constraints, or a mixture of them representing a spatial scene of a set of objects V.
>
> **Task:** Find a representation such that $\forall v_i, v_j \in V$, the qualitative relation (e.g. topological or directional relation) R_{ij} for v_i, v_j can be derived from the representation more efficiently than by direct geometric computation, while the storage space of the representation is much smaller than the QCN representation.

Several researches have considered the problem to reduce the storage of

qualitative relations. Fogliaroni [43] proposed a framework, called the spatial clustering index, to reduce the calculation and storage of qualitative relations, such that a qualitative information layer can be feasibly integrated into the current geographical databases. Al-Salman [1] studied a variant of the spatial clustering index that uses the clustering strategy DBSCAN [41] to better cluster the objects and uses the concave hull to approximate the shape of the objects in a cluster. In Chapters 5 and 6, we will discuss two different techniques for solving the compact representation problem, by making use of the existing minimum bounding boxes (MBRs) and the generated axis-aligned rectangles for spatial entities.

2.6 Summary

In this chapter, along with introduction to necessary concepts and notations, we have summarised several important tasks in the field of QSTR, including the consistency problem, the minimal labelling problem, the (weakly) global consistency problem, the redundancy problem, and the compact representation problem. We have also briefly reviewed the research on these problems and some of the most useful techniques in the field for solving them, including the two classical algorithms PC and PPC. The next chapters will present in detail how our research can help to better solve these problems.

Hopefully, with effective techniques for encoding and efficient techniques for reasoning on the information, in the future we will be able to build a comprehensive knowledge base containing both geometric and qualitative representation of spatio-temporal information. The knowledge base will become a useful interface for applications to handle qualitative information, and introduce a new and reliable user experience to explore our world.

Chapter 3

Distributive Subalgebras

3.1 Introduction

In the era of internet and smart electronic devices, the methods in QSTR need to deal with much larger datasets. In [70], Koubarakis et al. identified the demand of scalable algorithms in QSTR for large number of variables, to adapt for the data from Semantic Web. Researchers have devised more efficient algorithms that either make use of the property of specific subclasses of relations or exploit sparse structure of the constraint graphs.

By making use of sparse structure of constraint graphs, in recent years, Chmeiss and Condotta [18], as well as Sioutis and Koubarakis [112], observed that checking the consistency of IA and RCC8 QCNs can be accomplished more efficiently through the partial path consistency algorithm PPC. Amaneddine et al. [5] also devised a more efficient algorithm than PC to solve the minimal labelling problem of IA and RCC8 with the help of PPC on sparsely structured constraint graphs.

Besides, researchers have identified special subclasses where PC can be powerful enough to solve the minimal labelling problem or the global con-

sistency problem. For PA, van Beek [122] has shown that PC can solve the minimal labelling problem for QCNs over the subclass of convex PA relations (i.e. the relations in PA except $\neq$). For IA, Valdés-Pérez [120] showed that PC can solve the minimal labelling problem of complete and basic QCNs. Later, Bessière et al. [11] and Ligozat [81] identified the subclass $\mathcal{C}_{\text{IA}}$ of convex relations in IA that possesses this property. Amaneddine and Condotta [3] identified maximal and unique subclasses of IA such that PC can be used to directly solve both the minimal labelling problem and the global consistency problem, and later they [4] exploited these subclasses to solve the minimal labelling problem of the whole IA more efficiently. For RCC8, Chandra and Pujari [17] identified the subclass of convex RCC8 relations for which PC solves the minimal labelling problem.

However, these results are scattered and leave several important questions unanswered. For example, for specific subclasses of IA, PC was shown to be able to solve the global consistency problem. Do other calculi have subclasses with this property? Although PPC is more efficient than PC by exploiting the structure of QCNs, can it be further improved for relations with specific properties? In this chapter, we introduce the concept of *distributive subalgebras*, and show how it can help to answer these questions and improve the traditional techniques to solve the problems in QSTR.

The remainder of this chapter is organised as follows. In Section 3.2, we discuss some basic properties of qualitative calculi. In Section 3.3, we introduce the family of distributive subalgebras and give an interesting characterisation of it. Then we identify maximal distributive subalgebras of several widely used qualitative calculi in Section 3.4. After that, in Section 3.5, we show the reasoning power of path consistency on QCNs over distributive subalgebras and present some interesting properties of path consistent subnetworks

(i.e. a-closures) of such QCNs. In Section 3.6, we discuss the applicability of several more efficient algorithms than PC for QCNs over distributive subalgebras. In Section 3.7, we also give a further discussion about the connection of distributive subalgebras with conceptual neighbourhood graphs and finite domain CSPs. The last section concludes this chapter.

Bibliographic Note. The work reported in this chapter is based on the joint work with Sanjiang Li, Weiming Liu, and Michael Sioutis, which was first presented in the following publications.

- Zhiguo Long and Sanjiang Li: On Distributive Subalgebras of Qualitative Spatial and Temporal, COSIT, 2015, pp. 354–374.
- Zhiguo Long, Michael Sioutis, and Sanjiang Li: Efficient Path Consistency Algorithm for Large Qualitative Constraint Networks, IJCAI, 2016, pp 1202–1208.
- Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both: On Redundant Topological Constraints, Artificial Intelligence, 2015, vol. 225, pp. 51–76.

The results about the properties of distributive subalgebras are based on collaboration with Weiming Liu and Sanjiang Li. Weiming Liu first identified distributivity on the closure of basic RCC5 relations. I proved most of the properties of distributive subalgebras in general cases, while Sanjiang Li identified the connection to Helly's theorem and gave the proof for the sufficiency of Helly property for a subalgebra to be distributive (Theorem 3.8). With the algorithm (i.e. DPC) proposed by Sioutis et al. in [114] and the properties of distributive subalgebras, Michael Sioutis, Sanjiang Li, and I devised the algorithm (i.e. DPC+) which is more efficient than PPC. I proved its correctness, and analysed its performance in theory. Michael Sioutis performed the experiments in

the corresponding publication while the experiments in this thesis concerning the algorithm are conducted by myself.

3.2 Basic Properties of Qualitative Calculi

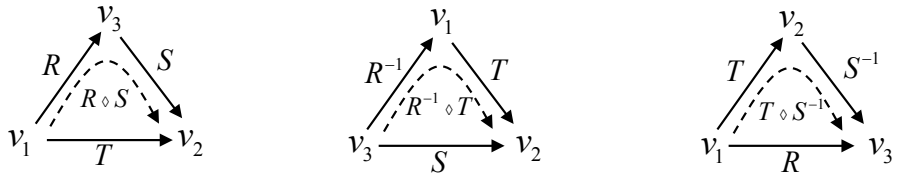
In the previous chapter, we have mentioned that while PA, IA, CRA and RA are all closed under the usual composition, the composition of two RCC5/8 relations is not necessarily a relation in RCC5/8 (cf. Section 2.4.1; [32, 78]). Therefore people introduced the concept of weak composition (cf. Section 2.4). We have the following relation between the usual composition and weak composition.

Proposition 3.1 ([84, 99])**.** *For $\mathcal{M}$ being PA, IA, CRA, or RA, weak composition is the same as composition, i.e. for any relations R and S in $\mathcal{M}$, we have $R \circ S = R \diamond S$.*

With the operations weak composition and converse $^{-1}$, and the identity relation $\mathsf{id}_{\mathcal{U}}$ as a basic relation (i.e. an atom), PA, IA, CRA, RA, and RCC5/8 are *relation algebras* [31, 34]. Then the weak composition and the converse of the relations in each of these calculi satisfy the following axioms.

- $\diamond$-associativity: $r \diamond (s \diamond t) = (r \diamond s) \diamond t$;
- $\diamond$-distributivity w.r.t. $\cup$: $(r \cup s) \diamond t = (r \diamond t) \cup (s \diamond t)$;
- $^{-1}$-distributivity: $(r \cup s)^{-1} = (r^{-1} \cup s^{-1})$;
- $^{-1}$-involutive distributivity: $(r \diamond s)^{-1} = (s^{-1} \diamond r^{-1})$.

More importantly, the calculi also satisfy the following Peircean Law.



$$(R \diamond S) \cap T \neq \varnothing \quad \Leftrightarrow \quad (R^{-1} \diamond T) \cap S \neq \varnothing \quad \Leftrightarrow \quad (T \diamond S^{-1}) \cap R \neq \varnothing.$$

Figure 3.1: Illustration of the Peircean Law.

Proposition 3.2 (See [31])**.** *For relations R, S, T in PA, IA, CRA, RA, and RCC5/8, we have the following* Peircean Law

$$(R \diamond S) \cap T \neq \varnothing \Leftrightarrow (R^{-1} \diamond T) \cap S \neq \varnothing \Leftrightarrow (T \diamond S^{-1}) \cap R \neq \varnothing. \tag{3.1}$$

Figure 3.1 gives an illustration of the configurations of relations in the Peircean Law. It can be considered as follows. Given a 3-variable QCN with constraints $(v_1 R v_3)$, $(v_3 S v_2)$, and $(v_1 T v_2)$, if, for example, for the tuple (v_1, v_3, v_2) we have $(R \diamond S) \cap T \neq \varnothing$, then for tuples (v_3, v_1, v_2) and (v_1, v_2, v_3) the corresponding intersections are also non-empty, i.e. $(R^{-1} \diamond T) \cap S \neq \varnothing$ and $(T \diamond S^{-1}) \cap R \neq \varnothing$.

In the literature, path consistency ($\mathbf{PC}$; cf. Definition 2.15) is a very useful local consistency of a QCN. The algorithm PC in the previous chapter (Algorithm 1) decides and establishes $\mathbf{PC}$ of a QCN. Although $\mathbf{PC}$ is just an approximation of the consistency, we have the following conclusion showing the reasoning power of $\mathbf{PC}$.

Proposition 3.3 ([6, 72, 82, 93, 100])**.** *For PA, IA, CRA, RA, and RCC5/8,* $\mathbf{PC}$ *is able to decide the consistency of scenarios (i.e. complete basic QCNs) and the consistency of QCNs over tractable subclasses containing all the basic relations.*

In later discussions, we assume that the qualitative calculus $\mathcal{M}$ have the

following properties:

$$\mathcal{M} \text{ is a relation algebra with } \diamond, {}^{-1} \text{ and } \mathsf{id}_U; \tag{3.2}$$

$$\text{Every path consistent scenario over } \mathcal{M} \text{ is consistent.} \tag{3.3}$$

In particular, the calculi PA, IA, CRA, RA and RCC5/8 satisfy these properties.

3.3 Distributive Subalgebra and Helly Property

The converse ($^{-1}$), weak composition ($\diamond$), and intersection ($\cap$) are three basic operations used in reasoning techniques of QSTR such as PC and PPC. These operations transform a QCN into another QCN, by changing the relations in the QCN. Some of the relations in the former QCN might no longer be in the same subclass as before, violating some useful properties (e.g. **PC** decides consistency) that are previously satisfied. Thus it would be more convenient to have a subclass of relations that are *stable* under these operations. A subalgebra is such a subclass. Note that a subclass $\mathcal{S}$ of a qualitative calculus $\mathcal{M}$ is called a *subalgebra* if $\mathcal{S}$ contains all the basic relations in $\mathcal{M}$ and is closed under converse, weak composition, and intersection.

Given a subclass of relations $\mathcal{X}$ of a qualitative calculus $\mathcal{M}$, we can generate a minimal subalgebra that contains $\mathcal{X}$, by calculating the closure of $\mathcal{X}$ under intersection, weak composition, and converse. Denote this subalgebra by $\widehat{\mathcal{X}}$, and in particular, $\widehat{\mathsf{B}}_{\mathcal{M}}$ denotes the closure of $\mathsf{B}_{\mathcal{M}}$ in $\mathcal{M}$.

Some subalgebras are *distributive* in the following sense.

Definition 3.4 (Distributive Subalgebra)**.** A subalgebra $\mathcal{S}$ is *distributive* if $R \diamond (S \cap T) = (R \diamond S) \cap (R \diamond T)$ and $(S \cap T) \diamond R = (S \diamond R) \cap (T \diamond R)$ for any R, S, $T \in \mathcal{S}$ with $S \cap T \neq \varnothing$.

As noted before, for $\mathcal{M}$ being one of PA, IA, CRA, RA, and RCC5/8, the closure of the basic relations in $\mathcal{M}$ is a distributive subalgebra, and every distributive subalgebra of $\mathcal{M}$ contains $\widehat{\mathsf{B}}_{\mathcal{M}}$ as a subclass.

In the following, we will give an useful characterisation of distributive subalgebras, i.e. the Helly Property.

3.3.1 Helly Property

Helly's theorem [26] is a very useful result in discrete geometry. For a collection of n intervals of $\mathbb{R}$, it says that if the intersection of any two of them is non-empty, then the intersection of the whole collection is also non-empty. As we will see later, relations in a distributive subalgebra have a similar property as the intervals of $\mathbb{R}$, i.e. the convex sets in the real line. Moreover, a subalgebra in which the relations have such property is a distributive subalgebra.

Definition 3.5. A subclass $\mathcal{S}$ of a qualitative calculus is called *Helly* if, for any finite n relations $R_1, \ldots, R_n \in \mathcal{S}$, we have

$$\bigcap_{i=1}^{n} R_i \neq \varnothing \quad \text{iff} \quad (\forall 1 \leq i \neq j \leq n)\ R_i \cap R_j \neq \varnothing. \tag{3.4}$$

This means that a subclass is Helly if, given that every two of n relations in the subclass have non-empty intersection, the n relations also have non-empty intersection.

The definition of a Helly subclass can be simplified from n relations to three relations, as the following proposition states. That is to say, a subclass is Helly if and only if, given that every two of three relations in the subclass have non-empty intersection, the three relations also have non-empty intersection.

Proposition 3.6. *A subclass $\mathcal{S}$ is Helly if and only if for any relations R, S, T*

$\in \mathcal{S}$, *we have*

$$R \cap S \cap T \neq \varnothing \quad \textit{iff} \quad R \cap S \neq \varnothing,\ R \cap T \neq \varnothing,\ S \cap T \neq \varnothing. \tag{3.5}$$

Proof. The "only if" part is straightforward. In the following we prove the "if" part.

Given any set of n relations $\mathcal{R} = \{R_i : 1 \leq i \leq n\}$ $(n \geq 1)$, assume that for any $1 \leq k \leq n-1$ and any subset of k relations $\{R_{i_j} \in \mathcal{R} : 1 \leq j \leq k, \}$, $R_{i_r} \cap R_{i_s} \neq \varnothing$ $(R_{i_r}, R_{i_s} \in \mathcal{R})$ iff $\bigcap_{j=1}^{k} R_{i_j} \neq \varnothing$. We want to show this also holds for $\mathcal{R}$. Now suppose that $R_i \cap R_j \neq \varnothing$ $(\forall 1 \leq i \neq j \leq n)$, then by assumption we have $\bigcap_{i=1}^{n-1} R_i \neq \varnothing$ and $\bigcap_{i=2}^{n} R_i \neq \varnothing$. Let $R = R_1$, $S = \bigcap_{i=2}^{n-1} R_i \neq \varnothing$, $T = R_n$, then we know $R \cap S = \bigcap_{i=1}^{n-1} R_i \neq \varnothing$ and $S \cap T = \bigcap_{i=2}^{n} R_i \neq \varnothing$, and $R \cap T \neq \varnothing$. Because by the "if" condition for any R, S, $T \in \mathcal{S}$ if $R \cap S \neq \varnothing$, $R \cap T \neq \varnothing$, and $S \cap T \neq \varnothing$, then $R \cap S \cap T \neq \varnothing$, we know that $\bigcap_{i=1}^{n} R_i \neq \varnothing$. Note that n is arbitrary, therefore $\mathcal{S}$ satisfies part of the definition of a Helly subclass, i.e. for any finite n relations $R_1, \ldots, R_n \in \mathcal{S}$, we have $\bigcap_{i=1}^{n} R_i \neq \varnothing$ *if* $R_i \cap R_j \neq \varnothing$ $(\forall 1 \leq i \neq j \leq n)$. It is straightforward forward to show that $\bigcap_{i=1}^{n} R_i \neq \varnothing$ *only if* $R_i \cap R_j \neq \varnothing$ $(\forall 1 \leq i \neq j \leq n)$. Hence, the "if" part is also proven. □

The result below shows that a distributive subalgebra is Helly, and such property is called the *Helly Property*.

Lemma 3.7 (Helly Property)**.** *Suppose $\mathcal{M}$ is a qualitative calculus that satisfies* (3.2)*, i.e. $\mathcal{M}$, with the weak composition, the converse operation, and the identity relation, is a relation algebra. Then every distributive subalgebra of $\mathcal{M}$ is Helly.*

Proof. To show a distributive subalgebra is Helly, by Proposition 3.6, we only need to consider the intersections of three arbitrary relations.

If $R \cap S \cap T \neq \varnothing$, then it is easy to see that $R \cap S \neq \varnothing$, $R \cap T \neq \varnothing$, and $S \cap T \neq \varnothing$. Next we show that if $R \cap S \neq \varnothing$, $R \cap T \neq \varnothing$, and $S \cap T \neq \varnothing$, then

$R \cap S \cap T \neq \varnothing$.

For two relations P and Q, we first note that $P \cap Q \neq \varnothing$ iff $\mathsf{id}_{\mathcal{U}} \in Q^{-1} \diamond P$. In fact, from $P \cap Q \neq \varnothing$, we know there exist two entities $a, b \in \mathcal{U}$ such that $(a, b) \in P \cap Q$. This implies that $(b, b) \in Q^{-1} \circ P$ as $(b, a) \in Q^{-1}$ and $(a, b) \in P$. Hence $\mathsf{id}_{\mathcal{U}} \cap Q^{-1} \circ P \neq \varnothing$. Note that $\mathsf{id}_{\mathcal{U}}$ is a basic relation in $\mathcal{M}$, then by the definition of weak composition, we have $\mathsf{id}_{\mathcal{U}} \in Q^{-1} \diamond P$. On the other hand, if $\mathsf{id}_{\mathcal{U}} \in Q^{-1} \diamond P$, then $\mathsf{id}_{\mathcal{U}} \cap Q^{-1} \circ P \neq \varnothing$. This implies that there exist two entities $a, b \in \mathcal{U}$ such that $(b, a) \in Q^{-1}$ and $(a, b) \in P$. Then $(a, b) \in P \cap Q$ and, hence, $P \cap Q \neq \varnothing$.

Assume on the contrary that $R \cap S \cap T = \varnothing$ and $R \cap S, R \cap T$ and $S \cap T$ are all non-empty. By the above observation, we have $\mathsf{id}_{\mathcal{U}} \in T^{-1} \diamond R$ and $\mathsf{id}_{\mathcal{U}} \in T^{-1} \diamond S$. Because R, S, T are relations in the distributive subalgebra $\mathcal{S}$ and $R \cap S \neq \varnothing$, we know

$$\mathsf{id}_{\mathcal{U}} \in (T^{-1} \diamond R) \cap (T^{-1} \diamond S) = T^{-1} \diamond (R \cap S).$$

Thus $(T^{-1} \diamond (R \cap S)) \cap \{\mathsf{id}_{\mathcal{U}}\} \neq \varnothing$ and, by the Peircean Law, $R \cap S \cap T \neq \varnothing$, which is a contradiction. Therefore, every distributive subalgebra of $\mathcal{M}$ is Helly. □

Interestingly, the above condition is also sufficient.

Theorem 3.8. *Suppose $\mathcal{M}$ is a qualitative calculus that satisfies* (3.2)*. Let $\mathcal{S}$ be a subalgebra of $\mathcal{M}$. Then $\mathcal{S}$ is distributive if and only if it is Helly.*

Proof. Since Lemma 3.7 has already shown the "only if" part, we only need to prove the "if" part. Suppose R, S, T are three relations in $\mathcal{S}$ and $S \cap T \neq \varnothing$. First, it is easy to see that $R \diamond (S \cap T) \subseteq (R \diamond S) \cap (R \diamond T)$. In fact, by the definition of weak composition, we have $R \diamond (S \cap T) = \bigcup\{r \diamond s : r \in R, s \in S \cap T, r, s \in \mathsf{B}_{\mathcal{M}}\}$. Note that $s \in S \cap T$ implies $s \in S$ and $s \in T$. Then for any $r \diamond s \subseteq R \diamond (S \cap T)$ with $r, s \in \mathsf{B}_{\mathcal{M}}$, $r \in R$ and $s \in S \cap T$, we have

$r \diamond s \subseteq (R \diamond S) \cap (R \diamond T)$. That is to say, $R \diamond (S \cap T) \subseteq (R \diamond S) \cap (R \diamond T)$. Next we show $R \diamond (S \cap T) \supseteq (R \diamond S) \cap (R \diamond T)$.

For any basic relation γ, by applying the Peircean Law twice, we have

$$\begin{aligned}
\gamma \notin R \diamond (S \cap T) &\Leftrightarrow \{\gamma\} \cap (R \diamond (S \cap T)) = \varnothing \\
\text{(Peircean Law)} &\Leftrightarrow (R^{-1} \diamond \gamma) \cap S \cap T = \varnothing \\
\text{(Helly)} &\Leftrightarrow (R^{-1} \diamond \gamma) \cap S = \varnothing \text{ or } (R^{-1} \diamond \gamma) \cap T = \varnothing \\
\text{(Peircean Law)} &\Leftrightarrow \{\gamma\} \cap (R \diamond S) = \varnothing \text{ or } \{\gamma\} \cap (R \diamond T) = \varnothing \\
\text{(Weak Composition)} &\Leftrightarrow \gamma \notin R \diamond S \text{ or } \gamma \notin R \diamond T.
\end{aligned}$$

This shows that $R \diamond (S \cap T) \supseteq (R \diamond S) \cap (R \diamond T)$ and thus $R \diamond (S \cap T) = (R \diamond S) \cap (R \diamond T)$. Similarly we can prove that $(S \cap T) \diamond R = (S \diamond R) \cap (T \diamond R)$. Therefore, $\mathcal{S}$ is Helly only if it is distributive. □

The Helly Property of distributive subalgebras is very useful and will be repeatedly used in later discussions.

3.4 Maximal Distributive Subalgebras

It would be interesting to see what a distributive subalgebra looks like for popular qualitative calculi. In this section, we will identify the maximal distributive subalgebras for PA, IA, RCC5/8, CRA, and RA. A distributive subalgebra $\mathcal{S}$ of a qualitative calculus $\mathcal{M}$ is *maximal* if there is no other distributive subalgebra of $\mathcal{M}$ that properly contains $\mathcal{S}$. It turns out that there are only a very limited number of maximal distributive subalgebras for each calculus and they have connections with subclasses defined in some other ways.

3.4.1 Maximal Distributive Subalgebras of PA, IA, and RCC5/8

Let $\mathcal{M}$ be PA, IA, RCC5, or RCC8 and $\mathcal{S}$ a subclass of $\mathcal{M}$. Recall that we write $\widehat{\mathcal{S}}$ for the subalgebra of $\mathcal{M}$ generated by $\mathcal{S}$ and write B for the set of basic relations in $\mathcal{M}$. To compute the maximal distributive subalgebras of $\mathcal{M}$, the general idea is to first compute $\widehat{\mathsf{B}}$, and then check by a program if $\widehat{\widehat{\mathsf{B}} \cup \mathcal{Z}}$ is distributive for some subset $\mathcal{Z}$ of $\mathcal{M}$. Algorithm 3 gives a more detailed procedure. The algorithm is practical for the task. In fact, for the hardest case, i.e. when $\mathcal{M}$ is IA, the execution on MATLAB[1] can finish in less than one hour.

Algorithm 3: MDS($\mathcal{M}$), an algorithm for finding maximal distributive subalgebras.

Input: A qualitative calculus $\mathcal{M}$ which can be PA, IA, RCC5 or RCC8, and $\mathcal{M}$ is considered as the set of relations in the calculus.
Output: The two maximal distributive subalgebras of $\mathcal{M}$.

```
1  R ← ∅;
2  foreach R ∈ M \ B̂ do
3  |  S_0 ← the subalgebra generated by B̂ ∪ {R};
4  |  if S_0 is distributive then
5  |  |  R ← R ∪ {R};
6  |  end
7  end
8  d ← an empty matrix to store d-relation information;
9  for each R ∈ R and each S ∈ R do
10 |  if the subalgebra generated by B̂ ∪ {R, S} is distributive then
11 |  |  d(R, S) ← true;
12 |  end
13 end
14 Find R_0, S_0 ∈ R s.t. d(R_0, S_0) ≠ true;
15 S_1 ← {R : d(R_0, R) = true};
16 S_2 ← {S : d(S_0, S) = true};
17 return the subalgebra generated by B̂ ∪ S_1 and the subalgebra generated by B̂ ∪ S_2;
```

[1] https://www.mathworks.com/products/matlab.html

In the algorithm, we first compute $\mathcal{R}$, the set of relations of R in $\mathcal{M} \setminus \widehat{\mathsf{B}}$ such that $\widehat{\widehat{\mathsf{B}} \cup \{R\}}$ is a distributive subalgebra. We then check for every pair of relations R, S in $\mathcal{R}$ if $\widehat{\widehat{\mathsf{B}} \cup \{R, S\}}$ is a distributive subalgebra. If this is the case, then we say that R is in d-*relation* with S and vice versa. Fortunately, the result shows that for PA, IA, and RCC5/8, there are precisely two subsets S_1 and S_2 s.t. $\mathcal{S}_1 \cap \mathcal{S}_2 = \varnothing$ and $\mathcal{S}_1 \cup \mathcal{S}_2 = \mathcal{R}$, which satisfy that each relation R in $\mathcal{S}_1$ ($\mathcal{S}_2$, respectively) is in d-relation with every other relation in $\mathcal{S}_1$ ($\mathcal{S}_2$, respectively), but is not in d-relation with any relation in $\mathcal{S}_2$ ($\mathcal{S}_1$, respectively). Moreover, $\widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_1}$ and $\widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_2}$ are both distributive subalgebras of $\mathcal{M}$. It can be seen that these are the only maximal distributive subalgebras of $\mathcal{M}$ as indicated by the following proposition.

Proposition 3.9. *Let $\mathcal{R}$, S_1, S_2 be as defined in Algorithm 3. Then $\widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_1}$ and $\widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_2}$ are the two unique maximal distributive subalgebras of PA, IA or RCC5/8.*

Proof. According to the verification, we know that $\widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_1}$ and $\widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_2}$ are distributive. Note that for any relation R in $\mathcal{M}$ that is not in $\mathcal{R}$, $\widehat{\widehat{\mathsf{B}} \cup \{R\}}$ is not distributive. Also, from each relation R in S_1 is not in d-relation with any relation S in S_2, we know that $\widehat{\widehat{\mathsf{B}} \cup \{R, S\}}$ is not distributive. Therefore, $\widehat{\widehat{\mathsf{B}} \cup \{R, S\}}$ is not a distributive subalgebra if $\{R, S\} \not\subseteq \widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_1}$ or $\{R, S\} \not\subseteq \widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_2}$. By this fact, adding any relation T that is not in $\widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_1}$ to it will make the resulting subalgebra become non-distributive. This is also the case for $\widehat{\widehat{\mathsf{B}} \cup \mathcal{S}_2}$. Then these two are maximal distributive subalgebras. The uniqueness can also be easily seen from the fact. □

In Appendix A, we list the maximal distributive subalgebras of PA, IA, RCC5 and RCC8.

3.4.2 Maximal Distributive Subalgebras of CRA

It should be noted that some CRA relations are not Cartesian products of some basic PA relations. In other words, it was not known before if maximal distributive subalgebras only consist of relations that are Cartesian products of PA relations or if they can be constructed from maximal distributive subalgebras of PA. Therefore, in the following, we first describe a procedure to compute maximal distributive subalgebras of CRA.

The procedure to compute the maximal distributive subalgebras of CRA is similar to Algorithm 3 for PA, IA, RCC5 and RCC8, but with some differences.

In detail, we first find out $\mathcal{R}$, the set of relations R in CRA such that $\widehat{\mathsf{B} \cup \{R\}}$ is distributive. There are eight different subalgebras in the set of subalgebras $\{\widehat{\mathsf{B} \cup \{R\}} : R \in \mathcal{D}\}$. We call these eight distributive subalgebras the *seed* subalgebras. Among these, only 4 are not contained in any other ones. We call these the *candidate* subalgebras. We then verify the following three facts:

1. For any pair of different candidate subalgebras $\mathcal{S}_i$ and $\mathcal{S}_j$, $\widehat{\mathcal{S}_i \cup \mathcal{S}_j}$ is not distributive.

2. For any pair of non-candidate subalgebras $\mathcal{S}_i$ and $\mathcal{S}_j$, $\widehat{\mathcal{S}_i \cup \mathcal{S}_j}$ is either a candidate subalgebra or not distributive.

3. For any pair of subalgebras $\mathcal{S}_i$ and $\mathcal{S}_j$ s.t. $\mathcal{S}_i$ is a candidate subalgebra, $\mathcal{S}_j$ is a non-candidate subalgebra, and $\mathcal{S}_j \not\subseteq \mathcal{S}_i$, we have that $\widehat{\mathcal{S}_i \cup \mathcal{S}_j}$ is not distributive.

Based upon the above facts, we show that the four candidate subalgebras are the only maximal distributive subalgebras of CRA.

To show maximality, suppose that $\mathcal{S}$ is one of the four candidate subalgebras. Let R be a relation in CRA which is not in $\mathcal{S}$. Then $\widehat{\mathcal{S} \cup \{R\}}$ is not

distributive. The reason is as follows. $\widehat{\mathsf{B} \cup \{R\}}$ is either a candidate subalgebra or a non-candidate subalgebra. For it is a candidate subalgebra, by the first fact, we know that $\widehat{\mathcal{S} \cup \{R\}}$ is not distributive (note that $\mathsf{B} \subseteq \mathcal{S}$). For it is a non-candidate subalgebra, note that $\widehat{\mathsf{B} \cup \{R\}} \not\subseteq \mathcal{S}$ since $R \notin \mathcal{S}$, then by the third fact $\widehat{\mathcal{S} \cup \{R\}}$ is still not distributive.

To show uniqueness, i.e. there are no other maximal distributive subalgebras, suppose that $\mathcal{S}'$ is a distributive subalgebra that is not a subset of any of the four candidate subalgebras. Then $\mathcal{S}'$ must contain at least two relations in $\mathcal{R}$, say R_1 and R_2. By the above facts, we know the closure of the union of $\widehat{\mathsf{B} \cup \{R_1\}}$ and $\widehat{\mathsf{B} \cup \{R_2\}}$ is either not distributive or one of the four candidate subalgebra. If it is the latter case, then $\mathcal{S}'$ would be either not distributive or a proper superset of one of the four candidate subalgebras. Note that the latter situation cannot happen as it contradicts the maximality of the four candidate subalgebras.

Interestingly, these four maximal distributive subalgebras of CRA exactly correspond to the Cartesian products of the maximal distributive subalgebras of PA, viz. $\mathcal{C}_{\mathrm{PA}} \otimes \mathcal{C}_{\mathrm{PA}}$, $\mathcal{C}_{\mathrm{PA}} \otimes \mathcal{S}_{\mathrm{PA}}$, $\mathcal{S}_{\mathrm{PA}} \otimes \mathcal{C}_{\mathrm{PA}}$, $\mathcal{S}_{\mathrm{PA}} \otimes \mathcal{S}_{\mathrm{PA}}$, where we interpret in a natural way a CRA relation e.g. $\{\mathsf{nw}, \mathsf{n}\}$ as $\{<, =\} \otimes \{>\}$.

3.4.3 Maximal Distributive Subalgebras of RA

Unlike the other small calculi we have discussed, RA has a large number (169) of basic relations, resulting in a total of 2^{169} relations in it. It becomes infeasible to exploit the former brute-force procedure to compute the maximal distributive subalgebras of RA. However, noting that the maximal distributive subalgebras of CRA are exactly the Cartesian products of the two maximal distributive subalgebras of PA, we conjectured that a similar situation happens to RA. This is indeed true.

Theorem 3.10. *RA has exactly four maximal distributive subalgebras, which are the Cartesian products of the two maximal distributive subalgebras of IA.*

Proof. The proof is given in Appendix B.1. □

In fact, the proof of the above theorem also applies to CRA and the extended version of RA, i.e. the Block Algebra (BA; [7]).

3.5 Path Consistency for QCNs over Distributive Subalgebras

Path consistency ($\mathbf{PC}$) can be used to decide the consistency of QCNs. Moreover, if $\mathcal{M}$ is a relation algebra and satisfies (3.3), i.e. every path consistent scenario is also *consistent*, then we can show that every path consistent QCN over a distributive subalgebra $\mathcal{S}$ of $\mathcal{M}$ is also weakly globally consistent and minimal. This specifically holds for distributive subalgebras of PA, IA, CRA, RA, and RCC5/8.

Theorem 3.11. *Let $\mathcal{M}$ be a qualitative calculus that satisfies* (3.2) *and* (3.3)*. Suppose $\mathcal{S}$ is a distributive subalgebra of $\mathcal{M}$. Then every path consistent QCN over $\mathcal{S}$ is weakly globally consistent and minimal.*

Proof. We defer the proof in Appendix B.1. □

Although the algorithm PC updates the relations in terms of "triangles", it in fact concerns about the restrictions made by the *paths*, when dealing with distributive subalgebras. Given a QCN over a qualitative calculus $\mathcal{M}$, a path π from a variable x to another variable y is a sequence of constraints $(c_1, \ldots, c_s)$ s.t. $c_i = (u_{i-1} R_i u_i)$, $u_0 = x$, and $u_s = y$. In this thesis, we use $\pi_{<i}$ to represent

the sub-path $(c_1, \ldots, c_{i-1})$ and $\pi_{>i}$ to represent the sub-path $(c_{i+1}, \ldots, c_s)$. Figure 3.2 illustrates a path $\pi = (c_1, c_2, \ldots, c_5)$ in a QCN, where $\pi_{<3} = (c_1, c_2)$ and $\pi_{>3} = (c_4, c_5)$. Note that a path can contain repeated variables. For example, the path (c_1, c_2, c_3, c_4, c') contains u_1 as a repeated variable. Specifically, when a path starts and ends at the same variable, e.g., the path (c_2, c_3, c_4, c'), we say such a path is a *cycle path*.

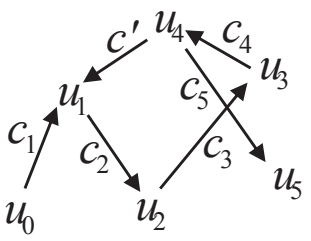


Figure 3.2: Illustration of path in a QCN.

The weak composition of a path $\pi = (c_1, \ldots, c_s)$, where $c_i = (u_{i-1} R_i u_i)$, is the relation in $\mathcal{M}$ defined as

$$\mathrm{CT}(\pi) \equiv R_1 \diamond R_2 \diamond \cdots \diamond R_s. \tag{3.6}$$

Since weak composition is associative, $\mathrm{CT}(\pi)$ is well defined. The length of a path π, denoted by $|\pi|$, is the number of constraints in it.

The following proposition shows some essence about why PC concerns about the paths for distributive subalgebras. Henceforth, we will use $\mathcal{P}_{xy}^{\mathcal{N}}$ to denote the set of all paths from x to y in a QCN $\mathcal{N}$.

Proposition 3.12. *Let $\mathcal{M}$ be one of PA, IA, CRA, RA, and RCC5/8, and $\mathcal{S}$ be a distributive subalgebra of $\mathcal{M}$. Suppose $\mathcal{N}$ is a consistent QCN over $\mathcal{S}$ and $\mathcal{N}_p$ its a-closure. Assume furthermore that (xSy) is a constraint in $\mathcal{N}_p$. Then $S = \bigcap_{\pi \in \mathcal{P}_{xy}^{\mathcal{N}}} \mathrm{CT}(\pi)$, i.e. S is the intersection of the weak compositions of all paths from x to y in $\mathcal{N}$.*

Proof. Suppose the QCN becomes stable in k steps of applying the update rule in (2.2) when enforcing $\mathbf{PC}$ by using the algorithm PC. For $l \leq k$, we write R_{ij}^l for the relation between v_i and v_j in the l-th step. We prove by using induction on l that every R_{ij}^l is the intersection of the weak compositions of *several* paths from v_i to v_j in $\mathcal{N}$.

When $l = 0$, this is clearly true. Suppose this is true for $l \leq s$. We show it also holds for $l = s + 1$. Suppose in this step the following updating rule is called

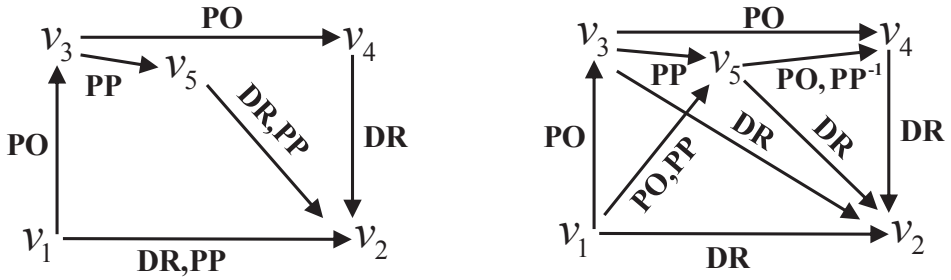
$$R_{ij}^{l+1} = (R_{ik}^l \diamond R_{kj}^l) \cap R_{ij}^l. \tag{3.7}$$

By induction hypothesis, we know R_{ij}^l is the intersection of the weak compositions of several paths from v_i to v_j in $\mathcal{N}$. Similar for R_{ik}^l and R_{kj}^l. Note that when joining a path from v_i to v_k and a path from v_k to v_j, we obtain a path from v_i to v_j. Because every constraint in $\mathcal{N}$ is taken from $\mathcal{S}$, in which weak composition distributes over non-empty intersections, it follows that $R_{ik}^l \diamond R_{kj}^l$ is identical to the intersection of the weak compositions of all these paths from v_i to v_j via v_k. It is now clear that R_{ij}^{l+1} also satisfies the property.

So far, we have shown for every constraint (xSy) in $\mathcal{N}_{\mathrm{p}}$ that S is the intersection of the weak compositions of *several* paths from x to y in $\mathcal{N}$. Because $\mathcal{N}_{\mathrm{p}}$ is path-consistent, the weak composition of *every* path from x to y in $\mathcal{N}_{\mathrm{p}}$ contains S. Therefore, S is also contained in the intersection of the weak compositions of *all* paths from x to y in $\mathcal{N}$. This shows that S is exactly the intersection of the weak compositions of *all* paths from x to y in $\mathcal{N}$. □

The requirement of $\mathcal{S}$ being distributive is necessary in the above lemma. Consider for example the consistent RCC5 QCN $\mathcal{N}$ over $\mathcal{H}_5$ shown in Figure 3.3. The intersection of the weak compositions of all paths from v_1 to

v_2 in $\mathcal{N}$ is $\{\mathbf{DR}, \mathbf{PP}\}$, while the relation that relates v_1 to v_2 in $\mathcal{N}_{\mathrm{p}}$ is $\{\mathbf{DR}\}$, which is strictly contained in $\{\mathbf{DR}, \mathbf{PP}\}$.



(a) A QCN $\mathcal{N}$ over $\mathcal{H}_5$. (b) The corresponding a-closure $\mathcal{N}_{\mathrm{p}}$.

Figure 3.3: An example showing that Proposition 3.12 does not generally hold for non-distributive subalgebras.

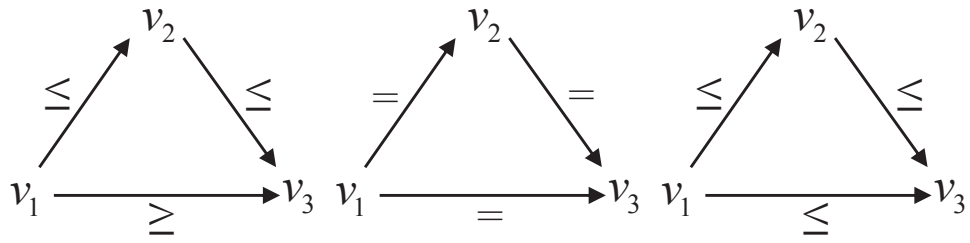
Some QCNs contain variables that are equal to some other variables. From a theoretical point of view, these variables can be removed, and sometimes we would only consider QCNs that do not have such variables. Formally, we call a QCN satisfying the following condition an *all-different* QCN.

$$(\forall i, j)[(i \neq j) \rightarrow (\mathcal{N} \not\models (v_i \,\mathsf{id}_{\mathcal{U}}\, v_j))]. \tag{3.8}$$

For example, the PA QCN $\mathcal{N}$ in Figure 3.4(a) is not an all-diffrent QCN, because its minimal subnetwork $\mathcal{N}_{\mathrm{m}}$ (and equivalently $\mathcal{N}$ itself) in Figure 3.4(b) entails, say, $(v_1 = v_2)$. On the other hand, $\mathcal{N}'$ in Figure 3.4(c) is all-different, because its minimal subnetwork $\mathcal{N}'_{\mathrm{m}}$ does not entail any constraint $(v_i = v_j)$.

By the proposition below, this restriction can be easily satisfied by distributive subalgebras as it is the same as requiring that the relation between two variables in the a-closure of $\mathcal{N}$ is not $\mathsf{id}_{\mathcal{U}}$ (e.g. **EQ** for RCC8).

Proposition 3.13. *Let $\mathcal{M}$ be one of PA, IA, CRA, RA, and RCC5/8, and $\mathcal{S}$ be a distributive subalgebra of $\mathcal{M}$. Suppose that $\mathcal{N} = \{v_i R_{ij} v_j : 1 \leq i, j \leq n\}$ is a*



(a) $\mathcal{N}$ (b) $\mathcal{N}_{\mathrm{m}}$ (c) $\mathcal{N}' = \mathcal{N}'_{\mathrm{m}}$

Figure 3.4: Illustration of all-different QCN.

consistent QCN over $\mathcal{S}$ and $\mathcal{N}_p$ its a-closure. Then, for any $i \neq j$, $\mathcal{N} \models (v_i \mathsf{id}_{\mathcal{U}} v_j)$ iff $(v_i \mathsf{id}_{\mathcal{U}} v_j)$ is in $\mathcal{N}_p$.

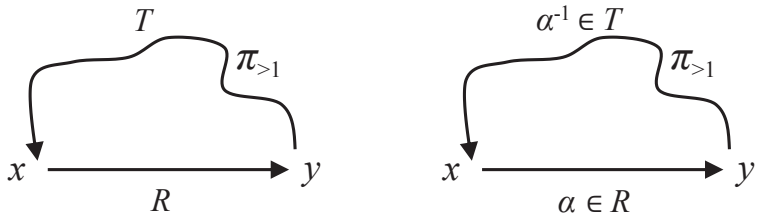
Proof. The sufficiency part is clear. For the necessity par, let S be the relation that relates v_i and v_j in $\mathcal{N}_{\mathrm{p}}$. Assume that $S \neq \mathsf{id}_{\mathcal{U}}$. By Theorem 3.11, S is the minimal label for v_i and v_j. Then $\exists \alpha \in S$ and $\alpha \neq \mathsf{id}_{\mathcal{U}}$ s.t. there is a solution of $\mathcal{N}_{\mathrm{p}}$ that satisfies the constraint $(v_i \alpha v_j)$ in $\mathcal{N}_{\mathrm{p}}$. Note that $\mathcal{N}_{\mathrm{p}}$ is equivalent to $\mathcal{N}$. Therefore, there is a solution of $\mathcal{N}$ that satisfies the constraint $(v_i \alpha v_j)$. This is a contradiction to that $\mathcal{N} \models (v_i \mathsf{id}_{\mathcal{U}} v_j)$. □

The observation below, which is about a special property of the weak composition of paths involving relations in one of PA, IA, CRA, RA, and RCC5/8, turns out to be useful in later proofs.

Lemma 3.14. *Suppose $\mathcal{N}$ is a consistent all-different QCN over $\mathcal{M}$ and $\pi = (c_1, c_2, ..., c_s)$ ($s \geq 2$) a path from x to itself (i.e. a cycle path) in $\mathcal{N}$ such that $c_i = (u_{i-1} R_i u_i)$, $u_0 = u_s = x$. Then $\mathrm{CT}(\pi)$ is the universal relation $\star$ if $\mathcal{M}$ is PA; contains $\mathsf{id}_{\mathcal{U}}$ if $\mathcal{M}$ is IA, CRA or RA; contains $\boldsymbol{O}_5 \equiv \{\boldsymbol{PO}, \boldsymbol{PP}, \boldsymbol{PP}^{-1}, \boldsymbol{EQ}\}$ if $\mathcal{M}$ is RCC5; contains $\boldsymbol{O}_8 \equiv \{\boldsymbol{PO}, \boldsymbol{TPP}, \boldsymbol{TPP}^{-1}, \boldsymbol{EQ}\}$ if $\mathcal{M}$ is RCC8.*

Proof. Write y for u_1. Let $R = R_1$ and $T = \mathrm{CT}(\pi_{>1}) = R_2 \diamond R_3 \diamond \ldots \diamond R_s$ (see Figure 3.5). Note that $y \neq x$ and $\pi_{>1}$ is a path from y to x. Suppose S is the relation from x to y in the a-closure of $\mathcal{N}$. Because $\mathcal{N}$ is consistent,

we know S is non-empty and $S \subseteq R$, $S \subseteq T^{-1}$. Furthermore, since $\mathcal{N}$ is all-different and hence satisfies (3.8), we know $S \neq \mathsf{id}_{\mathcal{U}}$. As a consequence, we know there is a basic relation $\alpha \neq \mathsf{id}_{\mathcal{U}}$ s.t. $\alpha \in S \subseteq R \cap T^{-1}$. Therefore, $\mathrm{CT}(\pi) = R \diamond T \supseteq \alpha \diamond \alpha^{-1}$. By checking the composition tables of PA, IA, CRA, RA, and RCC5/8, we can see that $\alpha \diamond \alpha^{-1}$ contains $\star$ for PA, contains $\mathsf{id}_{\mathcal{U}}$ for IA, CRA or RA, contains $\mathbf{O}_5$ for RCC5, and contains $\mathbf{O}_8$ for RCC8. □



(a) $R \cap T^{-1} \neq \varnothing$ (b) $\alpha \in R \cap T^{-1}$

Figure 3.5: Illustration for the proof of Lemma 3.14.

The following proposition gives a finer characterisation of a constraint (xSy) in $\mathcal{N}_{\mathrm{p}}$ than Proposition 3.12, in terms of paths in $\mathcal{N}$ that do not contain the constraint (xRy). Let W be the intersection of the weak compositions of all paths from x to y in $\mathcal{N} \setminus \{(xRy)\}$, i.e.

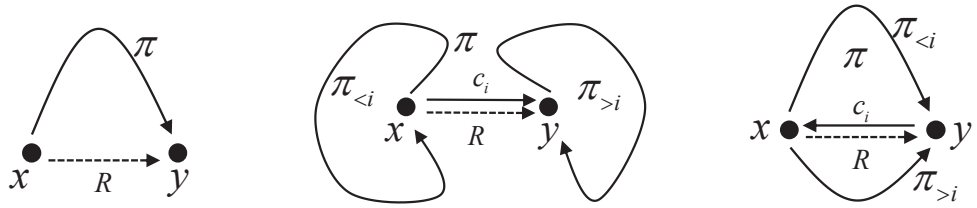
$$W = \bigcap \{\mathrm{CT}(\pi) : \pi \in \mathcal{P}_{xy}^{\mathcal{N} \setminus \{(xRy)\}}\}. \tag{3.9}$$

Proposition 3.15. *Let $\mathcal{S}$ be a distributive subalgebra of any one of the calculi PA, IA, CRA, RA, and RCC5/8. Suppose $\mathcal{N}$ is a consistent all-different QCN over $\mathcal{S}$ and $\mathcal{N}_{\mathrm{p}}$ its a-closure. Assume that (xRy) and (xSy) are the constraints from x to y in $\mathcal{N}$ and $\mathcal{N}_{\mathrm{p}}$ respectively. Then $S = R \cap W$.*

Proof. Because (xRy) is the only path with length 1 from x to y in $\mathcal{N}$, Proposition 3.12 in fact asserts that S is the intersection of R and $\bigcap\{\mathrm{CT}(\pi) : \pi \in$

$\mathcal{P}^{\mathcal{N}}_{xy}, |\pi| \geq 2\}$. Note that each path from x to y in $\mathcal{N} \setminus \{(xRy)\}$ is a path in $\mathcal{N}$ and has length ≥ 2. We know $S \subseteq R \cap W$.

To show that $S \supseteq R \cap W$, we only need to show that $\mathrm{CT}(\pi) \supseteq R \cap W$ for every path from x to y in $\mathcal{N}$ s.t. $|\pi| \geq 2$. Suppose that $\pi = (c_1, c_2, ..., c_s)$ $(s \geq 2)$ is such a path and $c_i = (u_{i-1} R_i u_i)$, $u_0 = x, u_s = y$.



(a) Case 1 (b) Case 2 (c) Case 3

Figure 3.6: Illustration of the three types of paths, where solid lines represent constraints or paths contained in π and the dashed line represents the constraint (xRy) in $\mathcal{N}$.

There are three types of paths (see Figure 3.6 for illustration).

Case 1. π contains neither (xRy) nor $(yR^{-1}x)$. Clearly π is a path from x to y in $\mathcal{N} \setminus \{(xRy)\}$. By the definition of W we have that $\mathrm{CT}(\pi) \supseteq W$.

Case 2. If $c_i = (xRy)$ for some $1 \leq i \leq s$, then $\mathrm{CT}(\pi) = \mathrm{CT}(\pi_{<i}) \diamond R \diamond \mathrm{CT}(\pi_{>i})$. Note that either $\pi_{<i}$ or $\pi_{>i}$ is a non-empty cycle path. By Lemma 3.14 we know the weak composition of each cycle path contains $\mathsf{id}_{\mathcal{U}}$. Therefore, we know $\mathrm{CT}(\pi) \supseteq R$.

Case 3. If $c_i = (yR^{-1}x)$ for some $1 \leq i \leq s$, then $\mathrm{CT}(\pi) = \mathrm{CT}(\pi_{<i}) \diamond \mathrm{CT}(\pi_{\geq i})$. Without loss of generality, we assume c_i is the first constraint in π such that $c_i = (yR^{-1}x)$. It is clear that $\pi_{<i}$ is a path of Case 1 and hence $W \subseteq \mathrm{CT}(\pi_{<i})$. Note that $\pi_{\geq i}$ is a path from y to itself. By Lemma 3.14 we know $\mathsf{id}_{\mathcal{U}} \in \mathrm{CT}(\pi_{\geq i})$ hence $\mathrm{CT}(\pi) = \mathrm{CT}(\pi_{<i}) \diamond \mathrm{CT}(\pi_{\geq i}) \supseteq W \diamond \mathsf{id}_{\mathcal{U}} = W$.

This shows that $R \cap W$ is contained in the weak composition of every path from x to y in $\mathcal{N}$ with length ≥ 2. Since S is the intersection of R and all paths

from x to y in $\mathcal{N}$ with length ≥ 2, this shows that $S \supseteq R \cap W$. Therefore we have $S = R \cap W$. □

As Proposition 3.12, the above result in general does not hold for non-distributive subalgebras. Consider again the QCN shown in Figure 3.3 and the constraint from v_1 to v_2. We have $R = \{\mathbf{DR}, \mathbf{PP}\}$, $S = \{\mathbf{DR}\}$, but $R \cap W = \{\mathbf{DR}, \mathbf{PP}\} \neq S$.

3.6 The Applicability of Algorithms Improving PC

The algorithm PC sometimes would be an overkill to decide the consistency of a QCN, especially when the QCN is sparsely structured. In the following, we present two alternatives of PC. Moreover, with the properties of distributive subalgebras discussed in the previous section, we will show their capability to solve the reasoning tasks including the consistency, the minimal labelling, and the (weakly) global consistency problems.

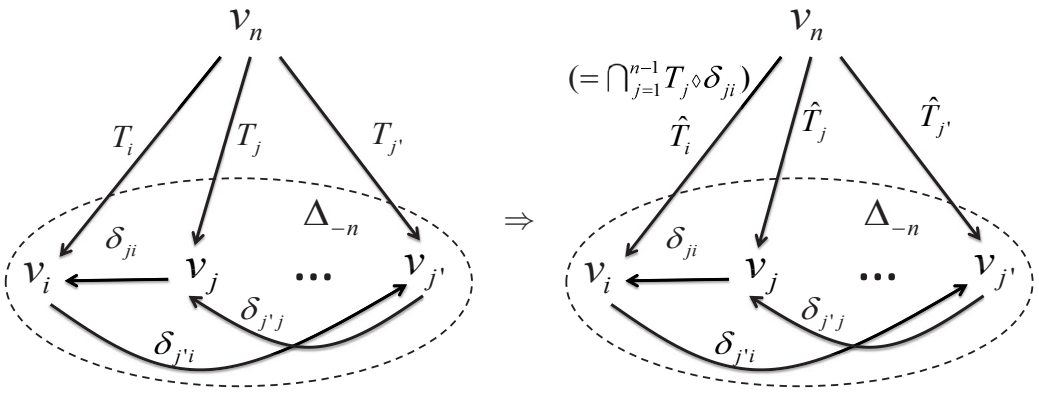
3.6.1 Variable Elimination

In [133], Zhang and Marisetti proposed a *variable elimination* method for solving the class of connected row convex (CRC) constraints [29, 123], which is a classical subclass in finite domain CSPs. The idea is to eliminate the variables from the constraint network one by one until a trivial problem is reached. Despite its simplicity, the algorithm is able to make use of the sparsity of the problem instances and significantly improves the efficiency to check consistency. One key property of CRC constraints is that any strong path consistent CRC constraint network is globally consistent. Recall that a similar property has been identified in Theorem 3.11 for QCNs over a distributive subalgebra.

The following lemma and theorem show that a similar variable elimination method applies to arbitrary QCN over any distributive subalgebra.

Lemma 3.16. *Let $\mathcal{M}$ be a qualitative calculus that satisfies* (3.2) *and* (3.3). *Suppose $\mathcal{N} = \{v_i R_{ij} v_j \ : \ 1 \leq i, j \leq n\}$ is a QCN over a distributive subalgebra $\mathcal{S}$ of $\mathcal{M}$ and $V = \{v_1, ..., v_n\}$. If $R_{ij} \subseteq R_{in} \diamond R_{nj}$ for every $1 \leq i, j < n$, then $\mathcal{N}|_{V_{-n}}$ is consistent only if $\mathcal{N}$ is consistent, where $\mathcal{N}|_{V_{-n}} = \{v_i R_{ij} v_j | 1 \leq i, j \leq n-1\}$ is the restriction of $\mathcal{N}$ on $V_{-n} = \{v_1, ..., v_{n-1}\}$.*

Proof. Suppose $\Delta_{-n} = \{\delta_{ij} \ : \ 1 \leq i, j < n\}$ (Figure 3.7(a)) is a consistent scenario of $\mathcal{N}|_{V_{-n}}$. We want to show Δ_{-n} can be extended to a consistent scenario of $\mathcal{N}$. To this end, we construct a path consistent refinement of $\mathcal{N}$ with the help of Δ_{-n}.



(a) (b)

Figure 3.7: Illustration of the proof of Lemma 3.16.

Write T_i for $R_{n,i}$ and let $\widehat{T}_i = \bigcap_{j=1}^{n-1} T_j \diamond \delta_{ji}$. It is easy to see that by replacing the constraints R_{ni} with $\widehat{T}_i$ and the constraints R_{ij} with δ_{ij} $(1 \leq i, j < n)$, the resulting QCN is a refinement of $\mathcal{N}$ (Figure 3.7(b)). To show this QCN is path consistent, we only need to show $\widehat{T}_j \subseteq \widehat{T}_i \diamond \delta_{ij}$ and $\widehat{T}_i \neq \varnothing$. The first statement

is true by noting that

$$\widehat{T}_i \diamond \delta_{ij} = (\bigcap_{j=1}^{n-1} T_{j'} \diamond \delta_{j'i}) \diamond \delta_{ij} = \bigcap_{j'=1}^{n-1} (T_{j'} \diamond \delta_{j'i} \diamond \delta_{ji}) \supseteq \bigcap_{j'=1}^{n-1} T_{j'} \diamond \delta_{j'j} = \widehat{T}_j.$$

To show $\widehat{T}_i \neq \varnothing$, because of the Helly Property of distributive subalgebras, i.e. the relations satisfy (3.5), we only need to show that $T_j \diamond \delta_{ji} \cap T_{j'} \diamond \delta_{j'i} \neq \varnothing$ for any $j \neq j'$. Applying the Peircean Law (see Proposition 3.2) twice, we have

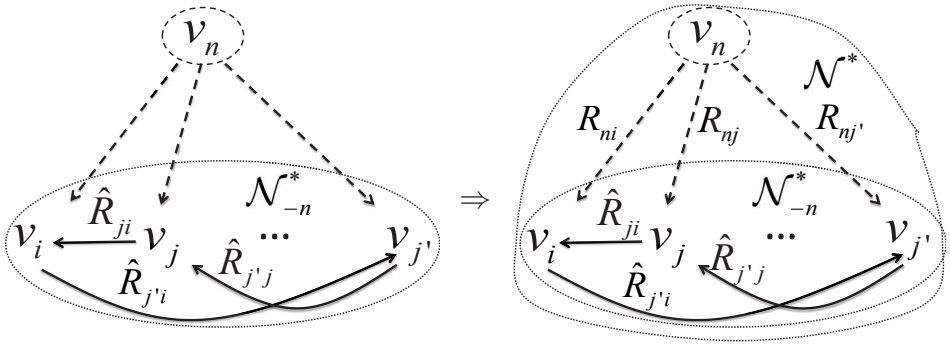
$$\begin{aligned} T_j \diamond \delta_{ji} \cap T_{j'} \diamond \delta_{j'i} \neq \varnothing \quad &\Leftrightarrow \quad T_{j'} \diamond \delta_{j'i} \diamond \delta_{ij} \cap T_j \neq \varnothing \\ &\Leftrightarrow \quad {T_{j'}}^{-1} \diamond T_j \cap \delta_{j'i} \diamond \delta_{ij} \neq \varnothing \\ &\Leftrightarrow \quad R_{j'n} \diamond R_{nj} \cap \delta_{j'i} \diamond \delta_{ij} \neq \varnothing. \end{aligned}$$

Because $\delta_{j'j} \subseteq R_{j'j} \subseteq R_{j'n} \diamond R_{nj}$ and $\delta_{j'j} \subseteq \delta_{j'i} \diamond \delta_{ij}$ (Δ_{-n} is consistent and hence path consistent), we have $R_{j'n} \diamond R_{nj} \cap \delta_{j'i} \diamond \delta_{ij} \neq \varnothing$ and $T_j \diamond \delta_{ji} \cap T_{j'} \diamond \delta_{j'i} \neq \varnothing$. □

From the above lemma, it will be easy to prove the following theorem.

Theorem 3.17. *Let $\mathcal{M}$ be a qualitative calculus that satisfies* (3.2) *and* (3.3). *Suppose $\mathcal{N} = \{v_i R_{ij} v_j \ : \ 1 \leq i, j \leq n\}$ is a QCN over a distributive subalgebra $\mathcal{S}$ of $\mathcal{M}$ and $V = \{v_1, ..., v_n\}$. Let $\mathcal{N}^*_{-n} = \{v_i \widehat{R}_{ij} v_j \mid \widehat{R}_{ij} = R_{ij} \cap (R_{in} \diamond R_{nj}), 1 \leq i, j \leq n-1\}$, i.e. $\mathcal{N}^*_{-n}$ is the QCN after eliminating v_n and updating the corresponding constraints of $\mathcal{N}$ (Figure 3.8(a)). Then $\mathcal{N}^*_{-n}$ is consistent if and only if $\mathcal{N}$ is consistent.*

Proof. Let $\mathcal{N}^* = \mathcal{N}^*_{-n} \cup \{v_i R_{in} v_n : 1 \leq i, j \leq n-1\}$ (Figure 3.8(b)). Then $\mathcal{N}^*$ is a refinement of $\mathcal{N}$ and if $\mathcal{N}^*$ is consistent then $\mathcal{N}$ is also consistent. Moreover, $\mathcal{N}^*_{-n}$ is the restriction of $\mathcal{N}^*$ on $\{v_1, \ldots, v_{n-1}\}$, and for every $1 \leq i, j < n$, $R_{ij} \subseteq R_{in} \diamond R_{nj}$. Then, by Lemma 3.16, we know that if $\mathcal{N}^*_{-n}$ is consistent



(a) $\mathcal{N}^*_{-n}$ (b) $\mathcal{N}^*$

Figure 3.8: Illustration of the proof of Theorem 3.17.

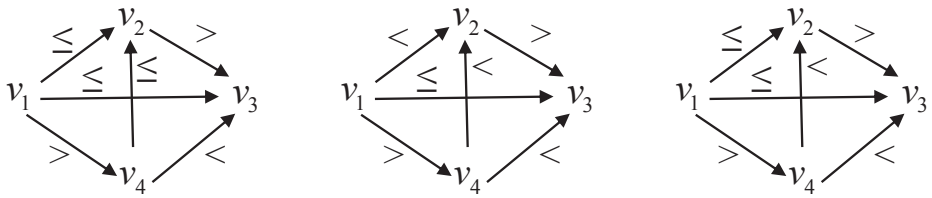
then $\mathcal{N}^*$ is also consistent. Therefore, $\mathcal{N}^*_{-n}$ is consistent only if $\mathcal{N}$ is consistent. Next, we show the "if" part.

Suppose $\Delta = \{v_i \delta_{ij} v_j : 1 \leq i, j \leq n\}$ is a consistent scenario of $\mathcal{N}$. We want to show that Δ is also a consistent scenario of $\mathcal{N}^*_{-n}$ when restricted to the variables $\{v_1, \ldots, v_{n-1}\}$. To this end, we only need to show that $\delta_{ij} \subseteq \widehat{R}_{ij}$ for any $1 \leq i, j \leq n-1$. Note that $\delta_{ij} \subseteq \delta_{in} \diamond \delta_{nj} \subseteq R_{in} \diamond R_{nj}$ and $\delta_{ij} \subseteq R_{ij}$, then $\delta_{ij} \subseteq \widehat{R}_{ij}$. Thus, Δ is also a consistent scenario of $\mathcal{N}^*_{-n}$ when restricted to the variables $\{v_1, \ldots, v_{n-1}\}$. □

Sioutis et al. [114] adopted the notion of *directional path consistency* and devised the algorithm DPC (Algorithm 4) that applies variable elimination to a QCN. The concept of directional path consistency first appeared in the context of CSPs from Dechter et al. [28].

Definition 3.18 (Directional Path Consistency; [114])**.** A QCN $\mathcal{N} = (V, \mathcal{C})$ is *directionally path consistent* (**DPC**) with respect to an ordering of its variables $\alpha = (v_1, \ldots, v_n)$ iff for all $v_i, v_k, v_j \in V$ with $i, j > k$ we have that $R_{ij} \subseteq R_{ik} \diamond R_{kj}$.

Figure 3.9(a) shows a PA QCN $\mathcal{N}$ that is not **DPC** w.r.t. the ordering $\alpha = (v_1, v_2, v_3, v_4)$, while $\mathcal{N}'$ in Figure 3.9(c) is **DPC** w.r.t. the ordering α. Note that



(a) $\mathcal{N}$ (b) $\mathcal{N}_{\mathrm{p}}$ (c) $\mathcal{N}'$

Figure 3.9: Illustration of directional path consistency, where $\mathcal{N}$ is not **DPC** and $\mathcal{N}'$ is the **DPC** subnetwork of $\mathcal{N}$ w.r.t. the ordering $\alpha = (v_1, v_2, v_3, v_4)$, which is not the same as $\mathcal{N}_{\mathrm{p}}$, the a-closure of $\mathcal{N}$.

$\mathcal{N}'$ is different from the a-closure of $\mathcal{N}$ shown in Figure 3.9(b) in this case. In fact, in $\mathcal{N}'$ we have $R_{23} = \{>\} \subseteq R_{21} \diamond R_{13} = \{\geq\} \diamond \{>\} = \{>\}$, which satisfies the **DPC** condition for v_1, v_2, and v_3, while $R_{12} = \{\leq\} \not\subseteq R_{13} \diamond R_{32} = \{<\}$, which means it is not **PC**.

The algorithm DPC applies variable elimination or, equivalently, achieves directional path consistency on a QCN. Note that the graph $G_{\mathcal{N}}$ is the constraint graph of $\mathcal{N}$ and when eliminating a variable, we do not need to consider all the edges in the completion of $G_{\mathcal{N}}$. This is different from PC, where all the edges in the complete graph will be considered. Also, DPC updates the relations in an ordered way, which saves much efforts, as we will see later in the analysis of its time complexity.

Based on the observation of variable elimination for QCNs over distributive subalgebras, in Theorem 3.17, Sioutis et al. [114] showed that DPC is sound and complete for deciding the consistency of QCNs over distributive subalgebras. Moreover, the resulting QCN $\mathcal{N}_e$ of DPC is directional path consistent, which we will call the **DPC** *subnetwork of* $\mathcal{N}$. This was first shown in [114].

Theorem 3.19 ([114]). *Let $\mathcal{N} = (V, \mathcal{C})$ be a QCN that is defined over a distributive subalgebra of a qualitative calculus that satisfies (3.2) and (3.3), and $\alpha = (v_1, \ldots, v_n)$ an ordering of V. Then, if $\mathcal{N}$ is consistent,* DPC *returns* $(\mathsf{True}, G, \mathcal{N}_e)$

Algorithm 4: DPC($\mathcal{N}$,α), a directional path consistency algorithm.

Input: A QCN $\mathcal{N} = (V, \mathcal{C})$ with n variables, and an ordering $\alpha = (v_1, \ldots, v_n)$.

Output: $(\mathsf{True}, G, \mathcal{N})$ or False.

```
1  E ← E(G_N);
2  G ← (V, E);
3  for v_k from v_1 to v_n do
4  |   F_k ← {v_s ∈ adj(v_k) : s > k};
5  |   foreach v_i, v_j ∈ F_k with i < j do
6  |   |   if {v_i, v_j} ∉ E then
7  |   |   |   E ← E ∪ {{v_i, v_j}};
8  |   |   end
9  |   |   temp ← R_ij ∩ (R_ik ⋄ R_kj);
10 |   |   if temp = ∅ then
11 |   |   |   return False;
12 |   |   end
13 |   |   if temp ⊂ R_ij then
14 |   |   |   R_ij ← temp;
15 |   |   |   R_ji ← temp^{-1};
16 |   |   end
17 |   end
18 end
19 return (True, G, N);
```

where $\mathcal{N}_e$ is the resulting directional path consistent QCN and returns False *otherwise.*

In fact, the resulting graph G in DPC is a *chordal* graph (cf. Definition 2.14), as observed in [114].

Proposition 3.20 ([114])**.** *Let $\mathcal{N} = (V, \mathcal{C})$ be a QCN that is defined over a distributive subalgebra of a qualitative calculus that satisfies (3.2) and (3.3), and $\alpha = (v_1, \ldots, v_n)$ an ordering of V. If* DPC *returns $(\mathsf{True}, G, \mathcal{N})$, then G is a chordal graph such that $G_{\mathcal{N}} \subseteq G$ and α is a perfect elimination ordering of G.*

Let us consider the time complexity of DPC. With the resulting chordal

graph G, the process of eliminating variables actually concerns the triangles in G. Suppose the ordering used in DPC is $\alpha = (v_1, \ldots, v_{n-1}, v_n)$. Then by Proposition 3.20, α is a perfect elimination ordering of the chordal graph G. Recall that $F_k = \{v_j \in \mathsf{adj}(v_k) : j > k\}$, and F_k induces a complete subgraph of G. For eliminating v_k, we have to update the relations between the variables in F_k. Therefore, the total number of updating operations is $\sum_{k=1}^{n} |F_k|(|F_k| - 1)/2$. Suppose the maximum vertex degree in G is d, then the total number of updating operations is bounded by $O(nd^2)$, where $n = |V|$. Because, we need to check each vertex and edge in G, the time complexity of DPC is $O(nd^2+n+m)$, where $m = |E|$. This has been observed by Sioutis et al. [114].

In terms of the number of triangles in G, we have another characterisation of the time complexity of DPC. With the above analysis, we know that each triangle in G is visited only once to update the relations. Thus, the time complexity of DPC is $O(t + n + m)$, where t is the number of triangles in G.

When $G_{\mathcal{N}}$ is sparsely structured, e.g., almost a tree, G will contain only a relatively small number of triangles, i.e. t is much smaller than $\binom{n}{3} = n(n-1)(n-2)/6$, the number of triangles in the complete graph. In this case, DPC is significantly more efficient than PC, which needs to visit all the triangles in the complete graph. Moreover, DPC will also have an advantage over the partial path consistency algorithm PPC (see Algorithm 2), which only visits the edges in the chordal graph rather than the complete graph. This is because PPC will visit each triangles more than once (e.g., as many times as $3|\mathsf{B}_{\mathcal{M}}|$). For the same reason, for cases where $G_{\mathcal{N}}$ is a very dense or complete graph, DPC will still have an advantage over PC.

In [114], Sioutis et al. have shown that both on randomly generated datasets and on real-world datasets that DPC can indeed significant improve the efficiency of checking consistency over both PPC and PC, and can scale to

very large datasets with tens of thousands of variables.

3.6.2 Partial Path Consistency

Although DPC is a significant improvement of PC to check consistency, it does not achieve $\mathbf{PC}$ on QCNs, and hence cannot solve the minimal labelling problem and the (weakly) globally consistency problem that require stronger reasoning power.

On the other hand, the algorithm PPC (Algorithm 2) achieves $\mathbf{PC}$ w.r.t. chordal graphs. Actually, in the context of finite domain CSPs for connected row convex constraint networks [29], Bliek and Sam-Haroud [13] demonstrated that the reasoning power of PPC is the same as PC on the common edges between a triangulation and the completion of the underlying constraint graph. However, for QCNs over general qualitative calculi, the reasoning power of PPC and PC on the common edges may not be identical (see e.g. [111]). Fortunately, for QCNs over a distributive subalgebra the result is affirmative. Sioutis et al. [113] first observed that.

Proposition 3.21 ([113])**.** *Let $\mathcal{M}$ be a qualitative calculus that satisfies* (3.2) *and* (3.3)*. Suppose $\mathcal{N} = \{v_i R_{ij} v_j : 1 \leq i, j \leq n\}$ is a QCN over a distributive subalgebra $\mathcal{S}$ of $\mathcal{M}$ and $V = \{v_1, ..., v_n\}$. Assume in addition that $G = (V, E)$ is a chordal graph such that $G_{\mathcal{N}} \subseteq G$. Then achieving $\mathbf{PPC}$ on G decides the consistency of $\mathcal{N}$ and results in the same relations on the edges of G as achieving $\mathbf{PC}$.*

Proof. The proof is given in Appendix B.1. □

By the above result, together with Theorem 3.11, we have the following conclusion.

Theorem 3.22. *Let $\mathcal{M}$ be a qualitative calculus that satisfies* (3.2) *and* (3.3). *Suppose $\mathcal{N} = \{v_i R_{ij} v_j \mid 1 \leq i, j \leq n\}$ is a QCN over a distributive subalgebra $\mathcal{S}$ of $\mathcal{M}$ and $V = \{v_1, ..., v_n\}$. Assume in addition that $G = (V, E)$ is a chordal graph such that $G_{\mathcal{N}} \subseteq G$ and $\mathcal{N}$ is* **PPC** *w.r.t. G. Then for any $\{v_i, v_j\} \in E$, R_{ij} is the minimal label between v_i and v_j.*

This observation means that PPC can be used as PC in some tasks that require the power of PC on the edges of the constraint graph. For instance, suppose we need to find out the minimal labels between several pairs of variables while the QCN is incomplete. If the QCN is over a distributive subalgebra, then we do not need to apply PC on the complete QCN, but just to add an edge between these pairs of variables, construct a corresponding chordal graph and apply PPC.

Next, let us have a look at the time complexity of PPC. Given a constraint network $\mathcal{N}$ and a chordal graph $G = (V, E)$ s.t. $G_{\mathcal{N}} \subseteq G$, regarding the triangles in $G = (V, E)$, PPC will visit each triangle no more than a constant times. Then PPC runs in time $O(t + n + m)$, where $n = |V|$, $m = |E|$, and t is the number of triangles in G. In terms of the maximum vertex degree d of G, PPC runs in time $O(n + md)$ where m is the number of edges in G (see, e.g., [13, 112]). When the graph G is a complete graph, PPC and PC are identical and both runs in $O(n^3)$ time. When the graph G is sparse, d and m, as well as t, will be much smaller than that of a complete graph, and hence in this case PPC is more efficient than PC. In fact, as noted in, e.g. [111], many real-world datasets induce sparsely structured (called scale-free) QCNs and PPC can be much more efficient than PC on these QCNs. Furthermore, in the following section, we will show that with the properties of distributive subalgebras, one can achieve **PPC** and hence **PC** more efficiently than using PPC and PC.

3.6.3 A More Efficient Algorithm to Achieve PPC and PC

In the previous two sections, we have seen that for distributive subalgebras PPC is useful to obtain the reasoning power of PC, while it is not as efficient as the weaker algorithm DPC, because it visits the triangles many more times than DPC. It is then natural to ask if we can make use of the efficiency of DPC while maintaining the reasoning power of PPC. In the context of Simple Temporal Problems [28], the answer is affirmative. Based on DPC, Planken et al. [96] proposed a new **PPC** enforcing algorithm, called P^{3}C, which only visits the triangles at most three times. Here, we propose a similar algorithm for QCNs over distributive subalgebras that is more efficient than the algorithm PPC.

Given a QCN $\mathcal{N}$ and an ordering $\alpha = (v_1, \ldots, v_n)$ of its variables, we first achieve **DPC** w.r.t. α on $\mathcal{N}$ using DPC. After this operation, we obtain a chordal graph $G = (V, E)$ with α as its perfect elimination ordering. Then, we update relations by iterating the variables in reverse order. In particular, for a variable v_k ($1 \leq k \leq n$), we consider the set F_k of variables that are adjacent to v_k and preceded by v_k in α, i.e. $F_k = \{v_j \in \mathsf{adj}(v_k) : j > k\}$. The relation between each $v_i \in F_k$ and v_k is updated with $\bigcap_{v_j \in F_k} R_{ij} \diamond R_{jk}$, where R_{ij} and R_{jk} are relations in the **DPC** subnetwork. For example, Figure 3.10 shows a QCN after achieving **DPC** w.r.t. the ordering $\alpha = (v_1, \ldots, v_5)$. In

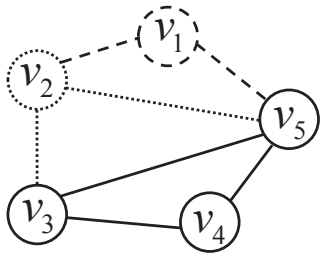


Figure 3.10: Illustration of updating relations by the for loop starting from Line 5 in the algorithm DPC+, where $v_k = v_2$ and $\alpha = (v_1, \ldots, v_5)$.

the figure, the variables surrounded by solid circle (i.e. v_5, v_4, v_3) have been processed by the updating procedure, the one by dotted circle (i.e. v_2) is being processed, and the one by broken circle (i.e. v_1) is going to be processed. The edges represented by solid line correspond to the relations that have been updated with $\bigcap_{v_j \in F_k} R_{ij} \diamond R_{jk}$, the edges by dotted line are being updated, and the edges by broken line are waiting to be updated. For this QCN, $F_k = F_2 = \{v_3, v_4\}$ and the relations R_{32} and R_{42} are being updated. The detailed steps are shown in Algorithm 5. We call this new algorithm DPC+.

Algorithm 5: DPC+($\mathcal{N}, \alpha$), a more efficient partial path consistency and path consistency algorithm

Input: A QCN $\mathcal{N} = (V, \mathcal{C})$ with n variables, and an ordering $\alpha = (v_1, \ldots, v_n)$ of V.
Output: True or False, a graph $G = (V, E)$, and an updated $\mathcal{N}$.

```
1  (result,G,N) ← DPC(N,α);
2  if result = False then
3  |  return (False,G,N);
4  end
5  for v_k from v_n to v_1 do
6  |  foreach v_i s.t. i > k and {v_i, v_k} ∈ E(G) do
7  |  |  F_k ← {v_j : j > k ∧ {v_j, v_k} ∈ E(G)};
8  |  |  R_ik ← ⋂_{v_j ∈ F_k} R_ij ◇ R_jk;
9  |  |  R_ki ← R_ik^{-1};
10 |  end
11 end
12 return (True,G,N);
```

The following theorem shows that DPC+ establishes $\mathbf{PPC}$ on a satisfiable QCN that is defined over a distributive subalgebra. Note that if we replace the graph G in Line 1 with the complete graph of the same order, DPC+ will achieve $\mathbf{PC}$.

Theorem 3.23. *Let $\mathcal{N} = (V, \mathcal{C})$ be a QCN that is defined over a distributive subal-*

gebra of a qualitative calculus that satisfies (3.2) and (3.3), and $\alpha = (v_1, \ldots, v_n)$ *an ordering of* V*. Then,* DPC+ *returns* $(\mathsf{True}, G, \mathcal{N}')$ *if and only if* $\mathcal{N}$ *is satisfiable, where* G *is a chordal graph such that* $G_{\mathcal{N}} \subseteq G$ *and* α *is a perfect elimination ordering of it, and* $\mathcal{N}'$ *is the* $\mathbf{PPC}$ *w.r.t.* G *subnetwork of* $\mathcal{N}$

Proof. We defer the proof to Appendix B.1. □

Interestingly, because DPC+ achieves $\mathbf{PPC}$ for distributive subalgebras, we know that DPC can be used to achieve minimality on the special pair of variables, i.e. (v_{n-1}, v_n). Formally, we have the following conclusion. Note that a similar circumstance has been observed in [96] for simple temporal problems.

Corollary 3.24. *Let* $\mathcal{N} = (V, \mathcal{C})$ *be a consistent QCN that is defined over a distributive subalgebra of a qualitative calculus that satisfies (3.2) and (3.3), and* $\alpha = (v_1, \ldots, v_n)$ *an ordering of* V*. Let* $\mathcal{N}'$ *be the* $\mathbf{DPC}$ *subnetwork of* $\mathcal{N}$ *w.r.t.* α*, and* $(v_{n-1} R'_{n-1,n} v_n)$ *the constraint between* v_{n-1} *and* v_n *in* $\mathcal{N}'$*. Then,* $R'_{n-1,n}$ *is the minimal label between* v_{n-1} *and* v_n*.*

Proof. Note that DPC+ establishes $\mathbf{PPC}$ on $\mathcal{N}$ and hence minimality on $\mathcal{N}$. From the procedure of $\mathcal{N}$, it is easy to see that after applying DPC on Line 1, the relation between v_{n-1} and v_n is not updated any more. Therefore, the resulting relation $R'_{n-1,n}$ by DPC for v_{n-1} and v_n is the same as the one in the minimal subnetwork of $\mathcal{N}$ after applying DPC+. This means $R'_{n-1,n}$ is minimal. □

The DPC+ algorithm only needs to update each triangle in a graph at most three times. This means that the time complexity of DPC+ is linear in the number of triangles in the graph.

Theorem 3.25. *Let* $\mathcal{N} = (V, \mathcal{C})$ *be a QCN, and* $\alpha = (v_1, \ldots, v_n)$ *an ordering of* V*. Then,* DPC+ *returns* $(\mathsf{True}, G, \mathcal{N}')$ *in* $\Theta(t + n + m)$ *time, or* $(\mathsf{False}, G, \mathcal{N}')$ *in* $O(t + n + m)$ *time, where* G *is a chordal graph w.r.t.* α *such that* $G_{\mathcal{N}} \subseteq G$*,* $\mathcal{N}'$ *is*

a subnetwork of $\mathcal{N}$, and t is the number of triangles in G and m is the number of edges in G.

Proof. The DPC algorithm considers each triangle in G exactly once. During this process, inconsistency can be detected. For each v_k and v_i such that $i > k$ and $\{v_i, v_k\} \in E$, lines 6–8 in DPC+ will consider all the triangles involving v_i and v_k once. Therefore, each triangle $\{v_i, v_j, v_k\}$ such that $i, j > k$ and $\{v_i, v_k\}, \{v_j, v_k\} \in E$ will be considered at most twice and, as such, due to iterating v_k from v_1 to v_n every triangle in the graph will be considered at most twice. Note that DPC+ also needs to scan through the vertices and edges in G, then the total time complexity of DPC+ is $\Theta(t + n + m)$. □

In terms of the maximum vertex degree d of G, with the above analysis, it is easy to see that DPC+ has a time complexity of $O(nd^2)$, where n is the number of variables.

While DPC+ only needs to visit each triangle in a graph at most three times, PPC usually requires to visit each such triangle many more times than that (e.g., $3|\mathsf{B}|$), as we have mentioned before. This is also the case with the **PC** enforcing algorithm PC, because when a complete graph is used in PPC, PPC is exactly the same as PC. Therefore, we expect DPC+ to be more efficient than both PPC and PC.

Empirical Evaluation of DPC+

In [87], we have shown that DPC+ is much more efficient than PPC (and hence PC) for establishing **PPC** on both random and real-world QCNs over distributive subalgebras of RCC8. Specifically, the random QCNs in [87] were generated by using the Barabási-Albert (BA) model, which is proposed by Barabási and Albert [8] and first exploited in QSTR by Sioutis et al. [111]. The BA

model BA(n, m) is usually used to generate *scale-free* graphs with n vertices. A graph is scale-free if the fraction $P(k)$ of vertices in the graph having k neighbouring vertices follows a power law, i.e. $P(k) \sim k^{-\gamma}$, where γ usually lies in the range $(2, 3)$ and may occasionally lie outside the range. The BA model BA(n, m) can generate scale-free graphs following $P(k) \sim k^{-3}$. Generally, it works as follows. Initially, there are m_0 vertices form a connected graph. The other vertices are added one by one, and m ($< m_0$) previously added vertices are randomly chosen to connect the newly added vertex with edges. m is thus called the *preferential attachment* value. When m is much smaller than n, the generated graphs are sparse, i.e. the number of edges of the graph is quite small compared to that of the corresponding complete graph with n variables. By associating qualitative relations with the edges, we obtain scale-free structured QCNs [111].

For these sparsely structured QCNs, the time and the number of (non-universal) constraint checks[2] needed for DPC+ to accomplish $\mathbf{PPC}$ are both significant smaller than that of PPC, especially when the number of variables becomes very large. Similar results were also observed on real-world datasets, e.g. for the administrative geography of Greece[3] with $1{,}732{,}999$ variables and $5{,}236{,}270$ constraints, DPC+ took 11.19s and checked about $36{,}000{,}000$ constraints, while PPC cost 398.63s and checked about $119{,}000{,}000$ constraints.

In the following, we show the advantage of DPC+ over PPC for establishing partial path consistency on QCNs over the distributive subalgebras of IA. In particular, our experiment is based on random consistent QCNs generated by the BA model BA(n,m). In [109] it was noted that $m = 3$ the generated QCNs lies in the phase transition region [101] of IA, where QCNs are equally

[2]A constraint check is performed when we compute the relation $r = R_{ij} \cap R_{ik} \diamond R_{kj}$ and check if $r \subset R_{ij}$.

[3]In: `http://www.linkedopendata.gr`

possible to be consistent or inconsistent. However, we found that for the distributive subalgebras of IA, this seems no longer the case. Even for $m = 2$, most generated QCNs are inconsistent if we randomly choose the relations in a distributive subalgebra with equal probability. Therefore, we use another technique[4] to select relations. First, we generate a scale-free graph by using BA(n, m). Then we construct a set of intervals by randomly selecting endpoints in a sequence of integers, i.e. 1 to 100. Note that there is a basic IA relation between any two intervals. For two variables that have an edge in the corresponding scale-free graph, we randomly select a relation from the distributive subalgebra of IA that contains the basic relation between these two variables. In this way, we obtain a consistent QCN that has scale-free structure and is over a distributive subalgebra of IA.

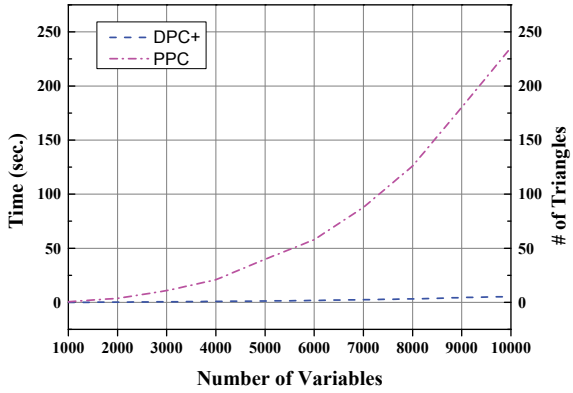
In our experiments, for each of the two maximal distributive subalgebras $\mathcal{C}_{\text{IA}}$ and $\mathcal{S}_{\text{IA}}$, and for each value of n from $1{,}000$ to $10{,}000$ with a step size of $1{,}000$, we generated 10 consistent QCNs with BA($n, m = 2$). We compared the performance of DPC+ and PPC on these datasets, where the implementations of both algorithms are based on the python implementations used in [87, 111]. We did not consider inconsistent instances in the experiment, because we want to focus on the performance of DPC+ to *establish* $\mathbf{PPC}$. If a QCN is inconsistent, $\mathbf{PPC}$ will not be established, and DPC+ becomes DPC, whose performance on detecting inconsistency has been analysed in [114]. In fact, on inconsistent QCNs over both $\mathcal{C}_{\text{IA}}$ and $\mathcal{S}_{\text{IA}}$ generated by BA(n,$m = 2$) for n from $1{,}000$ to $10{,}000$, we observed that DPC can detect inconsistency in 0.005s, while that of PPC is about 0.2s.

We used the maximum cardinality search algorithm [119] to obtain a variable elimination ordering for PPC to triangulate the graph and for DPC+ to

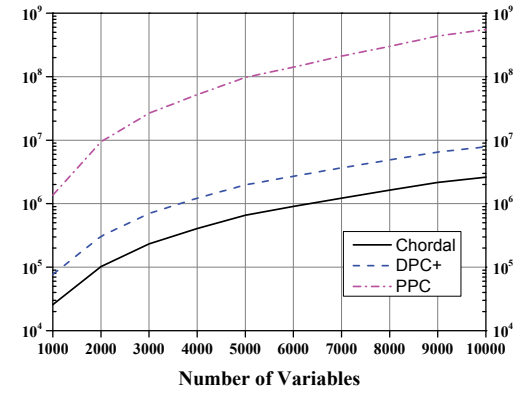
[4]From Michael Sioutis: `http://cgi.di.uoa.gr/~sioutis/work.php`

execute DPC. This algorithm can find a variable elimination ordering to triangulate a graph into a chordal graph $G = (V, E)$ in time $O(|V| + |E|)$.

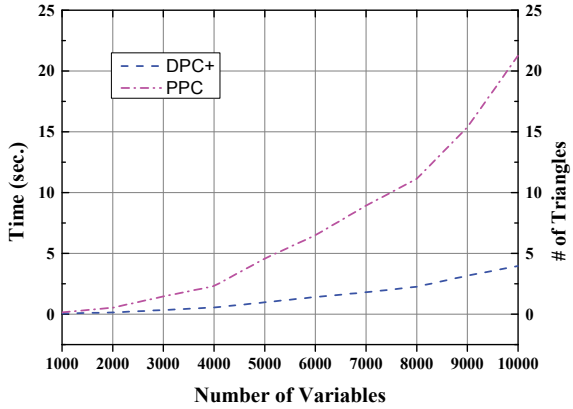
The performance was measured by the computation time and the number of visited triangles, on a computer running Red Hat Enterprise Linux 6.7 with a 2.9GHz Intel Xeon E5-2690 CPU, and 32GB 1600MHz RAM. We did not count the time for finding the ordering, as both PPC and DPC+ will use the same ordering and have the same time for finding the ordering.



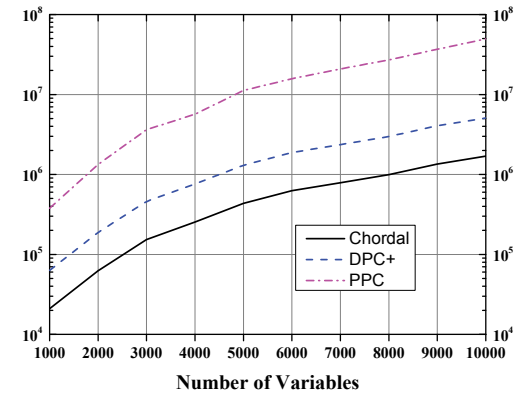
(a) Computation time for QCNs over $\mathcal{C}_{\text{IA}}$



(b) Number of visited triangles for QCNs over $\mathcal{C}_{\text{IA}}$, in logarithmic scale.



(c) Computation time for QCNs over $\mathcal{S}_{\text{IA}}$



(d) Number of visited triangles for QCNs over $\mathcal{S}_{\text{IA}}$, in logarithmic scale.

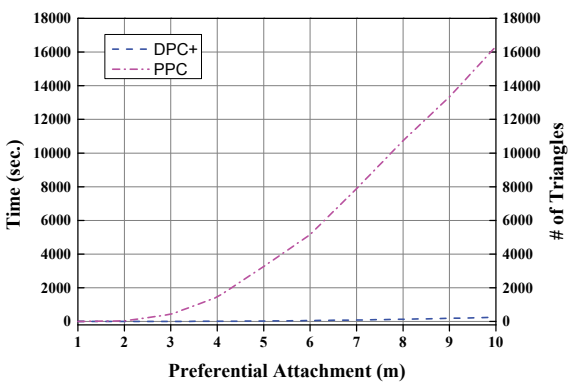
Figure 3.11: Comparison of DPC+ and PPC for increasing number of variables.

Figure 3.11 shows the results. From the results, we can see that DPC+

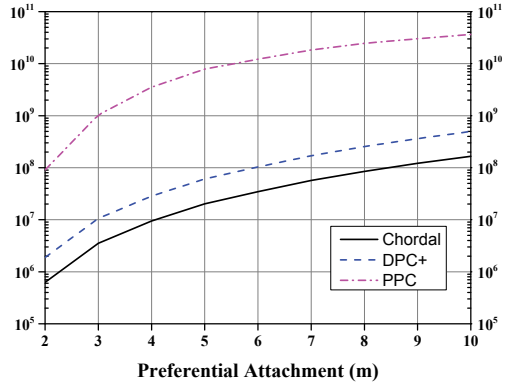
costs much less time than PPC and visits much less number of triangles for these QCNs. In particular, in Figure 3.11(a), when the number of variables grows to $10,000$, the computation time of PPC increases rapidly from about 1 second to more than 200 seconds, while the computation time of DPC+ remains below 10 seconds. Figure 3.11(b) shows the number of visited triangles by both algorithms, as well as that of the chordal graphs ("Chordal" in the figure). Note that the scale of the y-axis is now logarithmic. In the figure, the number of visited triangles by DPC+ is smaller than that by PPC for an order of magnitude. Moreover, the number of visited triangles by PPC grows faster than DPC+ as the number of variables and the number of triangles in the chordal graph increases. Similar phenomenon can be observed in Figure 3.11(c) and (d) for $\mathcal{S}_{\text{IA}}$. Note that the difference between DPC+ and PPC becomes smaller, which is probably due to that the compositions of the relations in this subclass are more likely to be restrictive and hence result in less propagation.

We also considered the effect of different values of m in the BA model BA(n, m). In particular, for $n = 5{,}000$, for each value of m in $\{1, 2, 3, \ldots, 10\}$, we generated 10 datasets for each of $\mathcal{C}_{\text{IA}}$ and $\mathcal{S}_{\text{IA}}$. The results are shown in Figure 3.12. Note that when $m = 1$ the generated QCNs are in tree structures that are already chordal and do not contain triangles. The results indicate that DPC+ is much more scalable than PPC when m increases. In particular, for the synthetic QCNs over $\mathcal{C}_{\text{IA}}$, the computation time of DPC+ remains very short (no more than 300 seconds) while that of PPC grows quickly up to more than $16{,}000$ seconds. For the synthetic QCNs over $\mathcal{S}_{\text{IA}}$, we can observe similar phenomenon. Also, we observed a larger growth rate of the number of visited triangles for PPC than that for DPC+. These results, together those of RCC8, suggest that DPC+ will be a more efficient substitute of the traditional algorithms for achieving $\mathbf{PPC}$ as well as $\mathbf{PC}$ for QCNs over distributive

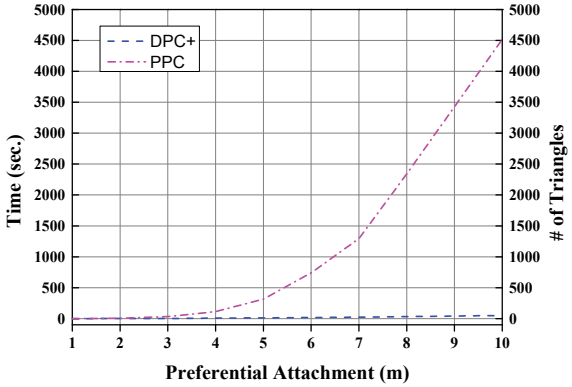
subalgebras.



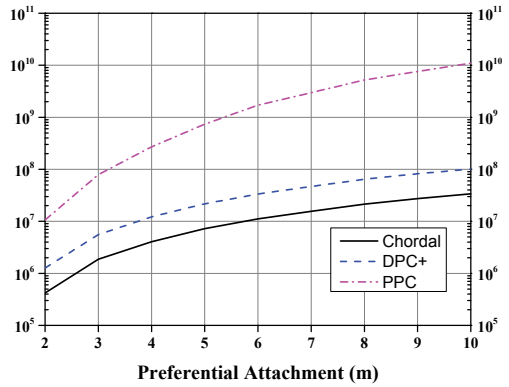
(a) Computation time for QCNs over $\mathcal{C}_{\text{IA}}$



(b) Number of visited triangles for QCNs over $\mathcal{C}_{\text{IA}}$, in logarithmic scale.



(c) Computation time for QCNs over $\mathcal{S}_{\text{IA}}$



(d) Number of visited triangles for QCNs over $\mathcal{S}_{\text{IA}}$, in logarithmic scale.

Figure 3.12: Comparison of DPC+ and PPC for increasing m in BA($n = 5000, m$).

3.7 Further Discussion

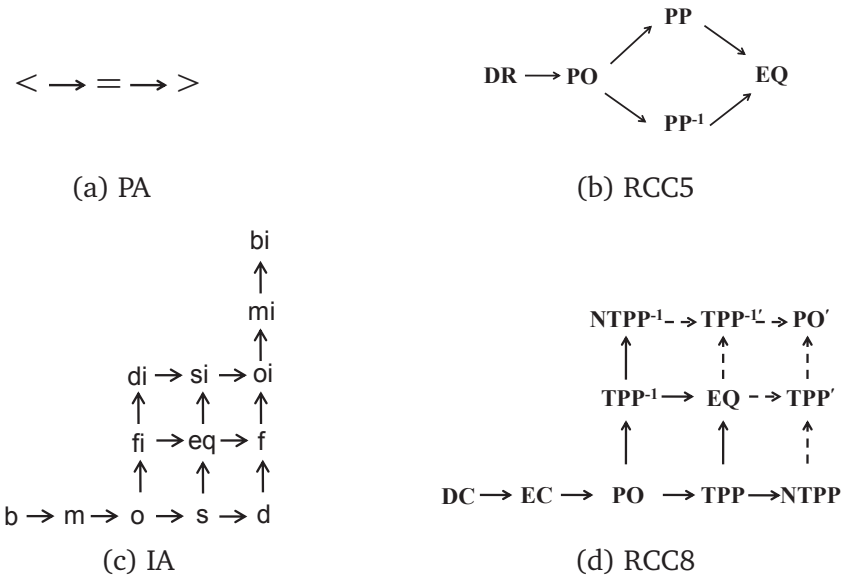
In this section we discuss the relation of distributive subalgebras with conceptual neighbourhood graphs [46] and classical CSPs.

3.7.1 Conceptual Neighbourhood Graph

Using the *conceptual neighbourhood graph* (CNG) of IA [46], Ligozat [81] gives a geometrical characterisation for the ORD-Horn relations $\mathcal{H}$. Consider the CNG of IA (shown in Figure 3.13 (b)) as a partially ordered set $(\mathsf{B}_{\mathrm{IA}}, \preceq)$ (by interpreting that any relation to be smaller than its right or upper neighbours). For $\theta_1, \theta_2 \in \mathsf{B}_{\mathrm{IA}}$ with $\theta_1 \preceq \theta_2$, we write $[\theta_1, \theta_2]$ as the set of basic interval relations θ such that $\theta_1 \preceq \theta \preceq \theta_2$, and call such a relation a *convex* interval relation. An IA relation R is called *preconvex* if it can be obtained from a convex relation by removing one or more basic relations with dimension lower than R, where the dimension of $\mathsf{b}, \mathsf{bi}, \mathsf{o}, \mathsf{oi}, \mathsf{d}, \mathsf{di}$ is 2, the dimension of $\mathsf{m}, \mathsf{mi}, \mathsf{s}, \mathsf{si}, \mathsf{f}, \mathsf{fi}$ is 1, and the dimension of eq is 0. For example, $[\mathsf{o}, \mathsf{eq}] = \{\mathsf{o}, \mathsf{s}, \mathsf{fi}, \mathsf{eq}\}$ is a convex relation and $\{\mathsf{o}, \mathsf{eq}\}$ is a preconvex relation. Ligozat has shown that ORD-Horn relations are precisely preconvex relations. Every path consistent QCN over $\mathcal{H}$ is consistent [93]. In addition, every path consistent QCN over the subclass of convex relations is globally consistent and minimal [81].

Interestingly, as we have seen before, the classes of convex IA and RCC8 relations are maximal distributive subalgebras of IA and RCC8 respectively. For IA, Ligozat characterises the convex relations by using the CNG of IA in Figure 3.13 (b). Similar idea applies to PA and RCC5 directly. For PA, the CNG is shown in Figure 3.13 (a). From the CNG of PA, we observe that the "convex" relations correspond to relations in $\mathcal{C}_{\mathrm{PA}} = \{<, =, >, \leq, \geq\}$, one of the maximal distributive subalgebras of PA. For RCC5, the CNG is shown in Figure 3.13 (b). The subclass of convex RCC5 relations is precisely the maximal distributive subalgebra $\mathcal{D}_{14}^5$.

The CNG of CRA is constructed by using the CNG of PA. For example, note that $<$ and $=$ are conceptual neighbours in the CNG of PA, and NW is defined as $x < x'$ and $y > y'$ and N is defined as $x = x'$ and $y > y'$. Then N and



(a) PA (b) RCC5

(c) IA (d) RCC8

Figure 3.13: Conceptual Neighbourhood Graphs.

NW should be conceptual neighbours in the CNG of CRA. The complete CNG of CRA is given in [82] and the subclass of convex CRA relations corresponds to the maximal distributive subalgebra $\mathcal{C}_{\mathrm{PA}} \otimes \mathcal{C}_{\mathrm{PA}}$. Like CRA, the CNG of RA is constructed by using the CNG of IA. The subclass of convex RA relations [6] is the maximal distributive subalgebra $\mathcal{C}_{\mathrm{IA}} \otimes \mathcal{C}_{\mathrm{IA}}$.

For RCC8, the situation is a little different. We need to revise the CNG by introducing three imaginary relations $\mathbf{TPP}', \mathbf{TPP}^{-1\prime}$ and $\mathbf{PO}'$ (see Figure 3.13, right). After this modification, Chandra and Pujari [17] identified the class of convex RCC8 relations, which is precisely the maximal distributive subalgebra $\mathcal{D}^8_{41}$.

At first, it seems that we can design an appropriate CNG, from which we can obtain the other maximal distributive subalgebras like the convex subclasses. However, this seems impossible, as for PA the maximal distributive subalgebra $\mathcal{S}_{\mathrm{PA}}$ contains $\{<, >\}$ but does not contain either $\leq$ or $\geq$.

3.7.2 Connection with Classical CSPs

For finite domain CSPs, Montanari [90] observed properties similar as the distributivity here. In particular, Montanari defined a concept called *star-distributive constraint network*, which is very similar to our notion of distributivity, except that it only requires the relations to form a closure w.r.t. the constraint network. If the concept of distributive subalgebra for finite domain CSPs is defined similar, then a constraint network over a distributive subalgebra is always star-distributive, while it is not clear whether a star-distributive network is always over a distributive subalgebra.

Table 3.1: An example showing that CRC relations are not always distributive.

$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$
R	S	T	$R \diamond (S \cap T)$	$R \diamond S \cap R \diamond T$

As we have seen, relations in a distributive subalgebra exhibit convexity in the sense of Helly. In finite CSP, row convex constraints [123] and (the more general) tree convex constraints [132] enjoy a similar property, which is specified w.r.t. the "rows" or "images" of the constraints rather than the constraints themselves. The relations R, S, and T in Table 3.1 are all CRC constraints but $R \diamond (S \cap T) \neq R \diamond S \cap R \diamond T$. Note that in this case $R \cap S \neq \varnothing$, $R \cap T \neq \varnothing$, $S \cap T \neq \varnothing$ but $R \cap S \cap T = \varnothing$. This means that CRC relations are not always distributive and do not always have the Helly Property in (3.5). It seems that the distributivity of finite domain constraints and the Helly Property are also somehow related.

3.8 Summary

In this chapter, we first discussed useful properties of distributive subalgebras and identified maximal distributive subalgebras for popular qualitative calculi. It turns out that several previously identified useful subclasses (e.g., the convex subclasses) coincide with some maximal distributive subalgebras. Based on the useful properties, for QCNs over distributive subalgebras, we showed the applicability of several efficient algorithms to solve the three important problems in QSTR, i.e. the consistency problem, the minimal labelling problem, and the (weakly) global consistency problem. The characterisation and properties of distributive subalgebras gives us insights of qualitative spatial and temporal reasoning, and the applicability of efficient algorithms will increase the scalability of a knowledge base and real-world applications for handling qualitative spatial and temporal information. In the next chapter, we will see that distributive subalgebras are also useful for solving the redundancy problem.

Chapter 4

Redundancy in QCNs

4.1 Introduction

Removing redundant information in QCNs is helpful for accelerating some tasks whose efficiency is strongly related to the density of the QCN or the number of constraints in the QCN. These include the adjustment of the position or the shape of the geometrical objects so that the spatial constraints between objects are satisfied [127, 128] and the comparison of QCNs by calculating the differences between paired relations between QCNs [53, 94], as well as dynamically updating QCNs. More details on this have been discussed in Section 2.5.4. Many other possible applications exist and are worth investigating in the future, such as how to efficiently merge two QCNs, how to reveal the essential structure of the QCN, and so on.

In the following, we will focus on discussing how distributive subalgebras can help to correctly and efficiently identify redundant constraints and prime subnetworks for QCNs. The general idea is as follows. We first investigate solving the redundancy problem of general QCNs in Section 4.2. Then in Section 4.3 we focus on discussing how distributive subalgebras can help to iden-

tify redundant constraints and prime subnetworks for QCNs, including results about unique prime subnetworks for some QCNs and the correspondence of redundant constraints in a QCN and its path consistent subnetwork. Section 4.4 discusses efficient algorithms to construct prime subnetworks.

Bibliographic Note. The work reported in this chapter is based on the joint work with Sanjiang Li, Weiming Liu, Matt Duckham, and Alan Both, which was first published in

- Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both: On Redundant Topological Constraints, Artificial Intelligence, 2015, vol. 225, pp. 51–76.

The work was initiated and motivated by Matt Duckham and Sanjiang Li. The first breakthrough was made by Weiming Liu, who found that $\hat{B}_5$, the closure of basic RCC5 relations, has the distributivity property. Through collaboration with Sanjiang Li and Weiming Liu, we proved more general results for distributive subalgebras of both RCC5 and RCC8, and devised the algorithm (i.e. Algorithm 6). I also constructed examples for illustrating that several results only hold for certain distributive subalgebras. Matt Duckham and Alan Both conducted the empirical evaluation in the publication, which is not included in this thesis.

4.2 Redundant Constraints

Recall that in Chapter 2, a QCN $\mathcal{N} = (V, \mathcal{C})$ entails a constraint $(v_i R_{ij} v_j)$, written as $\mathcal{N} \models (v_i R_{ij} v_j)$, if for every solution $(a_1, \ldots, a_n)$ of $\mathcal{N}$ we have $(a_i, a_j) \in R_{ij}$. Redundant constraints can be defined w.r.t. entailment.

Definition 4.1. Let $\mathcal{N}$ be a QCN over a qualitative calculus $\mathcal{M}$. A constraint $(v_i R v_j)$ in $\mathcal{N}$ is *redundant* if $\mathcal{N} \setminus \{(v_i R v_j)\} \models (v_i R v_j)$. A QCN is *prime* if it

does not have any redundant constraints, and *reducible* otherwise. If a QCN $\mathcal{N}'$ contains a subset of constraints of $\mathcal{N}$ is equivalent to $\mathcal{N}$ and is prime, then $\mathcal{N}'$ is a *prime subnetwork* of $\mathcal{N}$.

An example of redundant constraints and prime subnetwork has been given in Example 2.2.

The central question to find a prime subnetwork is to determine if a constraint is redundant. Generally, this is hard, as the following proposition shows.

Proposition 4.2. *Let $\mathcal{M}$ be one of IA, RCC5/8, CRA, and RA, and $\mathcal{N}$ be a QCN over $\mathcal{M}$. Suppose that (xRy) is a constraint in $\mathcal{N}$. It is then co-NP-complete to decide if (xRy) is redundant in $\mathcal{N}$.*

Proof. First of all, we note that (xRy) is redundant in $\mathcal{N}$ iff $(\mathcal{N} \setminus \{(xRy)\}) \cup \{xR^cy\}$ is inconsistent, where $R^c = \star \setminus R$, i.e. the complement of R. Since it is NP-complete to decide if a QCN over IA, RCC5/8, CRA, or RA is consistent, we know that the problem of determining if a constraint is redundant in a network is in co-NP. On the other hand, it is easy to construct a polynomial many-one reduction from the problem of checking the inconsistency of a QCN over one of these calculi to the redundancy problem, as shown in the following argument. Fix two variables x, y. Suppose $\mathcal{N}$ is an arbitrary QCN over one of these calculi with variables V and x, y are two variables in V. Then $\mathcal{N}$ is inconsistent iff $\mathcal{N} \setminus \{(xRy)\} \models (xR^cy)$ iff (xR^cy) is redundant in $(\mathcal{N} \setminus \{(xRy)\}) \cup \{xR^cy\}$. This shows that the redundancy problem is co-NP complete. □

By contrast, for PA, the problem of deciding if a constraint is redundant in a QCN is clearly in P.

To construct a prime subnetwork of a given QCN, a naive method would be to remove redundant constraints sequentially from the QCN until it is prime.

Suppose that $\{c_1, \ldots, c_k\}$ is the set of non-trivial redundant constraints in a QCN $\mathcal{N}$. For $0 \leq i \leq k-1$ and $\mathcal{N}_0 = \mathcal{N}$, define

$$\mathcal{N}_{i+1} = \begin{cases} \mathcal{N}_i \setminus \{c_{i+1}\} & \text{if } c_{i+1} \text{ is redundant in } \mathcal{N}_i; \\ \mathcal{N}_i & \text{if otherwise.} \end{cases} \tag{4.1}$$

Note that we have the following conclusion.

Proposition 4.3. *If a constraint (xRy) is not redundant in a QCN $\mathcal{N}$, then it is not redundant in any subset of $\mathcal{N}$ containing this constraint.*

Proof. Suppose $\mathcal{N}' \subseteq \mathcal{N}$ and (xRy) is a constraint in $\mathcal{N}'$. Assume on the contrary that (xRy) is redundant in $\mathcal{N}'$, i.e. $\mathcal{N}' \setminus \{(xRy)\} \models (xRy)$. Let $\mathcal{N}^* = \mathcal{N} \setminus \{(xRy)\}$ and $\mathcal{N}^{**} = \mathcal{N}' \setminus \{(xRy)\}$. Then any solution of $\mathcal{N}^{**}$ satisfies (xRy). Because $\mathcal{N}' \subseteq \mathcal{N}$, any solution of $\mathcal{N}^*$ is also a solution of $\mathcal{N}^{**}$ and satisfies (xRy). Therefore, (xRy) is redundant in $\mathcal{N}$, which is a contradiction. □

Then it is easy to see that $\mathcal{N}_k$ does not contain any redundant constraint and is equivalent to $\mathcal{N}$, i.e. $\mathcal{N}_k$ is a prime subnetwork of $\mathcal{N}$. Suppose that determining if a constraint is redundant in a QCN is in $O(t)$ time, then $\mathcal{N}_k$ can be obtained in $O(n^2 \cdot t)$ time. In particular, for $\mathcal{N}$ over PA, as van Beek [121] have shown that there is an algorithm to decide the consistency in $O(n^2)$ time, the redundancy of a constraint can also be decided in $O(n^2)$ time and a prime subnetwork can be constructed in $O(n^4)$ time. For the preconvex subclass of CRA, a prime subnetwork can also be constructed in $O(n^4)$ time, as Ligozat [82] showed that there is an algorithm to decide the consistency in $O(n^2)$ time. For IA, RCC5/8, and RA, by Proposition 3.3, the consistency of a QCN over a tractable subclass that contains all basic relations can be determined by the cubic time algorithm PC and hence a prime subnetwork can be constructed in $O(n^5)$ time for a QCN over a such tractable subclass. Here a tractable subclass

of RA specially refers to one of the currently known ones that contain all the basic relations (cf. [6]).

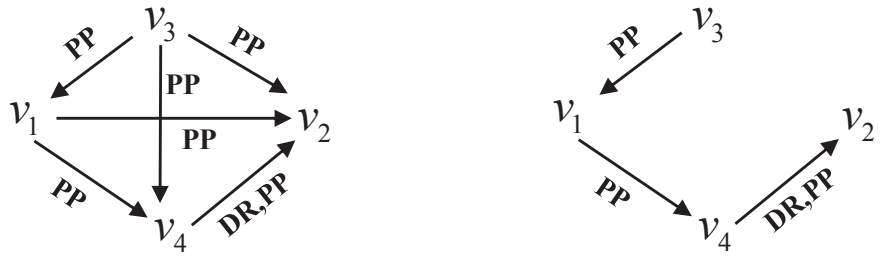
It would be interesting to specially consider the *core* of a QCN.

Definition 4.4. Let $\mathcal{N}$ is a QCN over a qualitative calculus $\mathcal{M}$. The core of $\mathcal{N}$, written as $\mathcal{N}_{\mathrm{c}}$, is the set of non-redundant constraints in $\mathcal{N}$.

Similar to the case of computing a prime subnetwork, by removing all the redundant constraints in $\mathcal{N}$, the core can be computed in $O(n^4)$ for the whole PA and the preconvex subclass of CRA, or $O(n^5)$ time for tractable subclasses of IA, RCC5/8, or RA that contain all the basic relations.

By Proposition 4.3, we know that the core $\mathcal{N}_{\mathrm{c}}$ is contained in every prime subnetwork of $\mathcal{N}$, and $\mathcal{N}_{\mathrm{c}}$ itself is prime. A natural question is whether $\mathcal{N}_{\mathrm{c}}$ is equivalent to $\mathcal{N}$ and hence it is a prime subnetwork of $\mathcal{N}$. In other words, we want to know if the prime subnetworks of $\mathcal{N}$ are unique. In general, however, this is not the case.

Example 4.1. Consider the QCN $\mathcal{N}$ over $\mathcal{H}_5$ shown in Figure 4.1(a). The core $\mathcal{N}_{\mathrm{c}}$ (shown in Figure 4.1(b)) is not equivalent to $\mathcal{N}$ and hence not a prime subnetwork of $\mathcal{N}$. This is because $(v_3\,\mathbf{DR}\,v_2)$ is feasible in $\mathcal{N}_{\mathrm{c}}$ but not in $\mathcal{N}$.



(a) a QCN $\mathcal{N}$ over $\mathcal{H}_5$ (b) the core $\mathcal{N}_{\mathrm{c}}$ of $\mathcal{N}$

Figure 4.1: An example showing that the core is not necessarily equivalent to the original QCN.

This means that the redundancy of some constraints might be dependent on other redundant constraints. Note that the relations in the above QCN are not all contained in a distributive subalgebra of RCC5. That is, this degenerate case is probably due to the absence of distributivity. On the other hand, for a QCN over a distributive subalgebra of PA or RCC5/8, if it is not *all-different* as specified in (3.8), then the core also might not be equivalent to $\mathcal{N}$.

Example 4.2. Suppose $\mathcal{N}$ is the QCN over RCC5 specified as below

$$\{v_1 \,\mathbf{P}\, v_2, v_2 \,\mathbf{P}\, v_3, v_3 \,\mathbf{P}\, v_1, v_1 \,\mathbf{PO}\, v_4, v_2 \,\mathbf{PO}\, v_4\},$$

where $\mathbf{P} = \{\mathbf{PP}, \mathbf{EQ}\}$. Then both **PO** constraints in $\mathcal{N}$ are redundant. This is because, by establishing PC on $\mathcal{N}$ we have that $v_1 \,\mathbf{EQ}\, v_2$, $v_1 \,\mathbf{EQ}\, v_3$, and $v_2 \,\mathbf{EQ}\, v_3$. Therefore, knowing one **PO** constraint will infer the other. $\mathcal{N}$ has no other redundant constraints and $\{v_1 \,\mathbf{P}\, v_2, v_2 \,\mathbf{P}\, v_3, v_3 \,\mathbf{P}\, v_2\}$ is the core of $\mathcal{N}$ but not equivalent to $\mathcal{N}$. It is easy to see that $\mathcal{N}_{\mathrm{c}} \cup \{v_1 \,\mathbf{PO}\, v_4\}$ and $\mathcal{N}_{\mathrm{c}} \cup \{v_2 \,\mathbf{PO}\, v_4\}$ are two prime subnetworks of $\mathcal{N}$.

This kind of circumstances could happen when $\mathcal{N}$ is not all-different. Nevertheless, in this case, we can easily merge the equal variables and update the constraint accordingly to make $\mathcal{N}$ all-different. In the following, we will assume that a QCN is all-different. Interestingly, with this assumption, a QCN $\mathcal{N}$ over any distributive subalgebra of PA or RCC5/8 has its core $\mathcal{N}_{\mathrm{c}}$ as the unique prime subnetwork, as we will see in the following section.

4.3 Unique Prime Subnetworks

We first summarise the results in Theorem 3.11 and in Theorem 3.22.

Theorem 4.5. *Let $\mathcal{S}$ be a distributive subalgebra of PA, IA, CRA, RA, and RCC5/8. Suppose $\mathcal{N}$ is a consistent QCN over $\mathcal{S}$ and $\mathcal{N}_p$ its a-closure. In addition, suppose $G = (V, E)$ is a chordal graph such that $G_{\mathcal{N}} \subseteq G$ and $\mathcal{N}_p^G$ is the* **PPC** *subnetwork of $\mathcal{N}$ w.r.t. G. Then $\mathcal{N}_p$ is identical to the minimal subnetwork of $\mathcal{N}$, and for any $(v_i, v_j) \in E$ and the constraint $(v_i T_{ij} v_j) \in \mathcal{N}_p^G$ we have that T_{ij} is the minimal label between v_i and v_j.*

Suppose $\mathcal{N}$ is a QCN over a distributive subalgebra $\mathcal{S}$ and $\mathcal{N}_{\mathrm{p}}$ its a-closure. Let (xRy) and (xSy) be the constraints from x to y in $\mathcal{N}$ and $\mathcal{N}_{\mathrm{p}}$ respectively. We will first show that, for distributive subalgebras of PA or RCC5/RCC8, (xRy) is redundant in $\mathcal{N}$ iff (xSy) is redundant in $\mathcal{N}_{\mathrm{p}}$. To this end, we first take a look at the following two lemmas showing that a constraint (xRy) in $\mathcal{N}$ is redundant iff R contains W, where

$$W = \bigcap \{\mathrm{CT}(\pi) : \pi \in \mathcal{P}_{xy}^{\mathcal{N} \setminus \{(xRy)\}}\}, \tag{4.2}$$

i.e. the intersection of the weak compositions of all paths from x to y in $\mathcal{N} \setminus \{(xRy)\}$. All of the following discussions assume that the QCNs are consistent.

Lemma 4.6. *Suppose $\mathcal{N}$ is a QCN over a qualitative calculus $\mathcal{M}$ and (xRy) a constraint in $\mathcal{N}$. Then (xRy) is redundant in $\mathcal{N}$ if $R \supseteq W$.*

Proof. Write $\mathcal{N}' = \mathcal{N} \setminus \{(xRy)\}$. For every path π from x to y in $\mathcal{N}'$, since (xRy) is redundant in $\mathcal{N}$, we know $\mathcal{N}'$ entails $(x\,\mathrm{CT}(\pi)y)$. By the definition of W, this implies that $\mathcal{N}'$ entails (xWy). Suppose $R \supseteq W$. It is clear that every solution of $\mathcal{N}'$ also satisfies (xRy), and therefore, (xRy) is redundant in $\mathcal{N}$. □

Note that this conclusion is true for any QCN over any qualitative calculus. Interestingly, for distributive subalgebras, the other way around is also true.

Lemma 4.7. *Let $\mathcal{S}$ be a distributive subalgebra of any one of the calculi PA, IA, RCC5/8, CRA and RA. Suppose $\mathcal{N}$ is an all-different QCN over $\mathcal{S}$ and (xRy) is a constraint in $\mathcal{N}$. Assume that W is defined as in* (4.2)*. Then (xRy) is redundant in $\mathcal{N}$ only if $R \supseteq W$.*

Proof. Suppose (xRy) is redundant in $\mathcal{N}$. Then each solution of $\mathcal{N}' = \mathcal{N} \setminus \{(xRy)$ also satisfies (xRy). Write (xTy) for the constraint between x and y in $\mathcal{N}'_p$, the a-closure of $\mathcal{N}'$. Note that the relation between x and y in $\mathcal{N}'$ is the universal relation $\star$. By Proposition 3.15 we know that $T = \star \cap W = W$. Furthermore, by Theorem 4.5, we know each basic relation in T is feasible in $\mathcal{N}'$, that is, for each basic relation in T, there is a solution of $\mathcal{N}'$ satisfying it. This implies that W $(= T)$ is contained in R. □

In general, this result does not hold for non-distributive subalgebras. For example, consider the QCN $\mathcal{N}$ over $\mathcal{H}_5$ shown in Figure 4.2 and the constraint $(v_1\mathbf{PP}v_2)$. It is easy to show that $\mathcal{N}$ is $\mathbf{PC}$, i.e., $\mathcal{N} = \mathcal{N}_{\mathrm{p}}$, and $(v_1\mathbf{PP}v_2)$ is redundant in $\mathcal{N}$. However, we have $W = \{\mathbf{PP}, \mathbf{EQ}\}$, which is not contained in $R = \mathbf{PP}$.

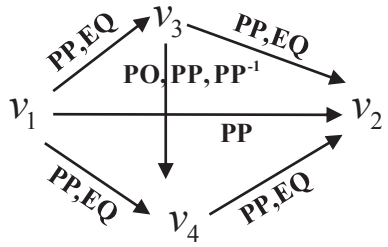


Figure 4.2: An example showing Lemma 4.7 generally does not hold for non-distributive subalgebras, where $\mathcal{N} = \mathcal{N}_{\mathrm{p}}$ is a path-consistent QCN over $\mathcal{H}_5$.

For distributive subalgebras, the above characterisation of redundant constraints can be made to paths of length two if $\mathcal{N}$ itself is path consistent.

Proposition 4.8. *Let $\mathcal{S}$ be a distributive subalgebra of any one of the calculi PA, IA, RCC5/8, CRA and RA. Suppose $\mathcal{N}$ is an all-different and path consistent QCN over $\mathcal{S}$. Then a constraint $(v_i S_{ij} v_j)$ is redundant in $\mathcal{N}$ iff $S_{ij} = \bigcap\{S_{ik} \diamond S_{kj} : k \neq i, j\}$, i.e., S_{ij} is the intersection of the weak compositions of all paths from v_i to v_j that have length two.*

Proof. By Lemma 4.6 we know that $(v_i S_{ij} v_j)$ is redundant in $\mathcal{N}$ if S_{ij} contains W_{ij}, the intersection of the weak compositions of all paths from v_i to v_j in $\mathcal{N} \setminus \{(v_i S_{ij} v_j)\}$. Then the "if" direction is clear. Next we prove the "only if" direction.

Suppose that $(v_i S_{ij} v_j)$ is redundant in $\mathcal{N}$. Let $\pi = (c_1, c_2, ..., c_s)$ ($s \geq 2$) be a path from v_i to v_j in $\mathcal{N} \setminus \{(v_i S_{ij} v_j)\}$ such that $c_t = (u_{t-1} S_t u_t)$ ($t = 1, \ldots, s$), $u_0 = v_i$ and $u_s = v_j$. We have $\mathrm{CT}(\pi) = S_1 \diamond \mathrm{CT}(\pi_{>1})$, where $\pi_{>1} = (c_2, \ldots, c_s)$. Suppose $u_1 = v_k$. Then $S_1 = S_{ik}$ and $\pi_{>1}$ is a path from v_k to v_j. For $\pi_{>1}$ containing the constraint $(v_i \star v_j)$, we have that $S_{kj} \subseteq \star = \mathrm{CT}(\pi_{>1})$. For $\pi_{>1}$ not containing the constraint $(v_i \star v_j)$, then $\pi_{>1}$ is a path in $\mathcal{N}$. Because $\mathcal{N}$ is $\mathbf{PC}$, we know by Proposition 3.12 that $S_{kj} \subseteq \mathrm{CT}(\pi_{>1})$. Therefore, $S_{ik} \diamond S_{kj} \subseteq \mathrm{CT}(\pi)$. Due to the arbitrariness of π, we know that $W_{ij} \supseteq \bigcap_{k \neq i,j} S_{ik} \diamond S_{kj}$. Since $(v_i S_{ij} v_j)$ is redundant in $\mathcal{N}$, by Lemma 4.7, $S_{ij} \supseteq W_{ij} \supseteq \bigcap_{k \neq i,j} S_{ik} \diamond S_{kj}$. On the other hand, by the fact that $\mathcal{N}$ is $\mathbf{PC}$, we have $\bigcap_{k \neq i,j} S_{ik} \diamond S_{kj} \supseteq S_{ij}$. Therefore $S_{ij} = \bigcap_{k \neq i,j} S_{ik} \diamond S_{kj}$. □

Still, this result does not hold in general for non-distributive subalgebras. Consider again the $\mathbf{PC}$ QCN $\mathcal{N}$ over RCC5 over $\mathcal{H}_5$ shown in Figure 4.2. Although $(v_1 \,\mathbf{PP}\, v_2)$ is redundant in $\mathcal{N}$, $S_{13} \diamond S_{32} \cap S_{14} \diamond S_{42} = \{\mathbf{PP}, \mathbf{EQ}\}$ strictly contains $\{\mathbf{PP}\}$.

From the above proposition, we have another characterisation of the redundant constraints in the a-closure $\mathcal{N}_{\mathrm{p}}$ w.r.t. the paths in the original QCN $\mathcal{N}$.

Corollary 4.9. *Let $\mathcal{N}$ is an all-different QCN over any distributive subalgebra of the calculi PA, IA, RCC5/8, CRA and RA. Suppose $\mathcal{N}_p$ is the a-closure of $\mathcal{N}$ and (xSy) is a constraint in $\mathcal{N}_p$. Assume that (xSy) is redundant in $\mathcal{N}_p$. Then $S = \bigcap\{\mathrm{CT}(\pi) : \pi \in \mathcal{P}^{\mathcal{N}}_{xy}, |\pi| \geq 2\}$, i.e. the intersection of all paths from x to y in $\mathcal{N}$ with length ≥ 2.*

Proof. Because (xSy) is redundant in $\mathcal{N}_{\mathrm{p}}$, by Proposition 4.8, we know $S = \bigcap\{S_{xk} \diamond S_{ky} : k \neq x, y\}$, i.e. the intersection of the weak compositions of all paths with length 2 from x to y in $\mathcal{N}_{\mathrm{p}}$. For each constraint $(u_i S_{ij} u_j)$ in such a path, Proposition 3.12 shows that $S_{ij} = \bigcap_{\pi \in \mathcal{P}^{\mathcal{N}}_{ij}} \mathrm{CT}(\pi)$, i.e. the intersection of the weak compositions of all paths from u_i to u_j in $\mathcal{N}$. Replace each S_{ij} in $\bigcap\{S_{xk} \diamond S_{ky} : k \neq x, y\}$ with $\bigcap_{\pi \in \mathcal{P}^{\mathcal{N}}_{ij}} \mathrm{CT}(\pi)$. By the distributivity of weak composition, we know that $S = \bigcap\{S_{xk} \diamond S_{ky} : k \neq x, y\}$ becomes the intersection of the weak compositions of *several* paths from x to y in $\mathcal{N}$ with length ≥ 2, and $S \supseteq \bigcap\{\mathrm{CT}(\pi) : \pi \in \mathcal{P}^{\mathcal{N}}_{xy}, |\pi| \geq 2\}$, the intersection of the weak compositions of *all* the paths from x to y in $\mathcal{N}$ with length ≥ 2. By Proposition 3.12 again we know that $\forall \pi$ that is a path from x to y in $\mathcal{N}$, $S \subseteq \mathrm{CT}(\pi)$ and hence $S \subseteq \bigcap_{\pi \in \mathcal{P}^{\mathcal{N}}_{xy}} \mathrm{CT}(\pi)$. Note that $\bigcap_{\pi \in \mathcal{P}^{\mathcal{N}}_{xy}} \mathrm{CT}(\pi) \subseteq \bigcap\{\mathrm{CT}(\pi) : \pi \in \mathcal{P}^{\mathcal{N}}_{xy}, |\pi| \geq 2\}$. This shows that $S = \bigcap\{\mathrm{CT}(\pi) : \pi \in \mathcal{P}^{\mathcal{N}}_{xy}, |\pi| \geq 2\}$. □

We next show that (xRy) is redundant in $\mathcal{N}$ iff (xSy) is redundant in $\mathcal{N}_{\mathrm{p}}$ for some distributive subalgebras. First we have the following observation.

Lemma 4.10. *Suppose $\mathcal{N}_1$ and $\mathcal{N}_2$ are two equivalent QCNs and $\mathcal{N}_2$ is a refinement of $\mathcal{N}_1$. Assume further that $(x\ R_1\ y) \in \mathcal{N}_1$ and $(x\ R_2\ y) \in \mathcal{N}_2$ are two corresponding constraints between x and y. Then $(x\ R_1\ y)$ is redundant in $\mathcal{N}_1$ only if $(x\ R_2\ y)$ is redundant in $\mathcal{N}_2$.*

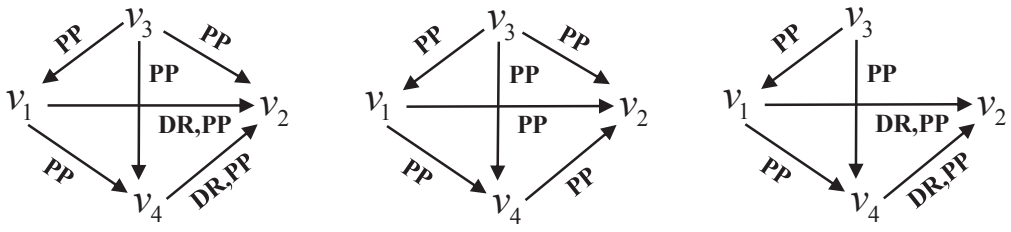
Proof. Write $\mathcal{N}'$ for $\mathcal{N}_1 \setminus \{(xR_1y)\}$ and $\mathcal{N}''$ for $\mathcal{N}_2 \setminus \{(xR_2y)\}$. Suppose (xR_1y) is redundant in $\mathcal{N}_1$. Then $\mathcal{N}' \models (xR_1y)$. Because $\mathcal{N}''$ is a refinement of $\mathcal{N}'$,

we know that every solution of $\mathcal{N}''$ is a solution of $\mathcal{N}'$ and hence also satisfies (xR_1y). Therefore, every solution of $\mathcal{N}''$ is a solution of $\mathcal{N}_1$. Note that $\mathcal{N}_1$ and $\mathcal{N}_2$ are equivalent. Then every solution of $\mathcal{N}''$ is also a solution of $\mathcal{N}_2$ and hence satisfies (xR_2y). Thus, (xR_2y) is redundant in $\mathcal{N}_2$. □

Proposition 4.11. *Let $\mathcal{S}$ be a distributive subalgebra of PA or RCC5/8. Suppose $\mathcal{N}$ is an all-different QCN over $\mathcal{S}$. Assume that (xRy) and (xSy) are the constraints from x to y in $\mathcal{N}$ and $\mathcal{N}_p$ respectively. Then (xRy) is redundant in $\mathcal{N}$ iff (xSy) is redundant in $\mathcal{N}_p$.*

Proof. For the necessary part, note that $\mathcal{N}_\mathrm{p}$ refines $\mathcal{N}$ and $\mathcal{N}_\mathrm{p}$ is equivalent to $\mathcal{N}$, then by Lemma 4.10, we have that (xRy) is redundant in $\mathcal{N}$ only if (xSy) is redundant in $\mathcal{N}_\mathrm{p}$. The proof of the sufficiency part is given in Appendix B.2. □

The result also does not hold in general for non-distributive subalgebras. Consider the QCN $\mathcal{N}$ over $\mathcal{H}_5$ shown in Figure 4.3 and the constraint from v_3 to v_2. It is clear that the constraint $(v_3\mathbf{PP}v_2)$ is redundant in $\mathcal{N}_\mathrm{p}$. However, $(v_3\mathbf{PP}v_2)$ is not redundant in $\mathcal{N}$. This is because $(v_3\,\mathbf{DR}\,v_2)$ is consistent with $\mathcal{N} \setminus \{(v_3\mathbf{PP}v_2)\}$ (shown in Figure 4.3(c)). Actually, it is easy to construct a solution $\{a_1, a_2, a_3, a_4\}$ of $\mathcal{N} \setminus \{(v_3\mathbf{PP}v_2)\}$, in which $(a_3\,\mathbf{PP}\,a_1)$, $(a_1\,\mathbf{PP}\,a_4)$ and $(a_2\,\mathbf{DR}\,a_j)$ for $j = 1, 3, 4$.

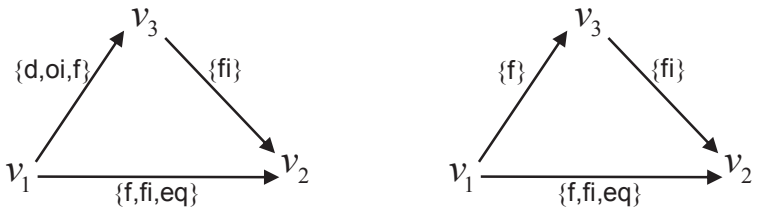


(a) a QCN $\mathcal{N}$ over $\mathcal{H}_5$ (b) its a-closure $\mathcal{N}_\mathrm{p}$ (c) $\mathcal{N} \setminus \{(v_3\mathbf{PP}v_2)\}$

Figure 4.3: An example showing Proposition 4.11 generally does not hold for non-distributive subalgebras.

Moreover, for the two maximal distributive subalgebras of IA, the fact that (xSy) is redundant in $\mathcal{N}_{\mathrm{p}}$ also does not necessarily imply that (xRy) is redundant in $\mathcal{N}$. Actually, we can find an example showing this even for QCN over $\widehat{\mathsf{B}}_{\mathrm{IA}}$.

Example 4.3. In the QCN in Figure 4.4(a), $x\{\mathsf{f}, \mathsf{fi}, \mathsf{eq}\}y$ is not redundant because replacing it with $x\{\mathsf{si}\}y$ still retains the consistency of the QCN. But in the corresponding a-closure in Figure 4.4(b), $x\{\mathsf{f}, \mathsf{fi}, \mathsf{eq}\}y$ is redundant, since $\mathsf{f} \diamond \mathsf{fi} = \{\mathsf{f}, \mathsf{fi}, \mathsf{eq}\}$. We should note that a similar example for QCNs over RA can be easily constructed by using the relations in the form $R \otimes \mathsf{id}_{\mathcal{U}}$.

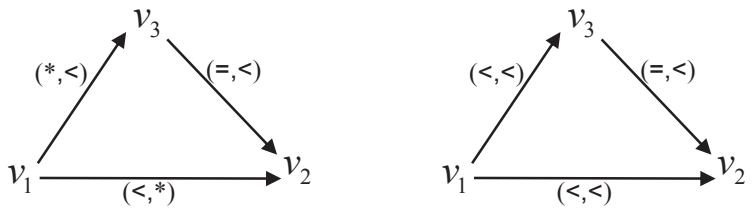


(a) The original IA QCN. (b) The corresponding a-closure.

Figure 4.4: An example showing Proposition 4.11 does not necessarily hold for IA QCN over $\widehat{\mathsf{B}}_{\mathrm{IA}}$, where $x\{\mathsf{f}, \mathsf{fi}, \mathsf{eq}\}y$ is not redundant in the original QCN but redundant in the a-closure.

In addition, for $\widehat{\mathsf{B}}_{\mathrm{CRA}}$ of CRA, Proposition 4.11 is also not true, as shown by the following example.

Example 4.4. In Figure 4.5, the constraint $v_1(<, \star)v_2$ is not redundant in the original QCN, because removing it yields $v_1(\star, <)v_2$ in the a-closure. On the other hand, the constraint $v_1(<, <)v_2$ is redundant in the a-closure of the original QCN. We note that this is due to a "partial" violation of the all-different condition, as the minimal relation between v_1 and v_2 would be $v_1(<, =)v_2$ where the two variables "equal" in the last dimension.



(a) The original CRA QCN. (b) The corresponding a-closure.

Figure 4.5: An example showing Proposition 4.11 does not necessarily hold for CRA QCN over $\widehat{\mathsf{B}}_{\mathrm{CRA}}$, where $(v_1(<,*)v_2)$ is not redundant in the original QCN but $(v_1(<,<)v_2)$ is redundant in the a-closure.

Recall that Theorem 4.5 asserts that $\mathcal{N}_{\mathrm{p}}$ is minimal. Proposition 4.11 can be rephrased as follows:

Proposition 4.12. *Let $\mathcal{S}$ be a distributive subalgebra of PA or RCC5/8. Suppose $\mathcal{N}$ is an all-different QCN over $\mathcal{S}$ and $\mathcal{N}_m$ the minimal subnetwork of $\mathcal{N}$. Assume that (xRy) and (xSy) are the constraints from x to y in $\mathcal{N}$ and $\mathcal{N}_m$ respectively. Then (xRy) is redundant in $\mathcal{N}$ iff (xSy) is redundant in $\mathcal{N}_m$.*

With the above results, by the theorem below, we know the core is also the unique prime subnetwork of an all-different QCN $\mathcal{N}$ over any distributive subalgebra of PA or RCC5/8.

Theorem 4.13. *Let $\mathcal{S}$ be a distributive subalgebra of PA or RCC5/8. Suppose $\mathcal{N}$ is an all-different QCN over $\mathcal{S}$ and $\mathcal{N}_{\mathrm{c}}$ the core of $\mathcal{N}$. Then $\mathcal{N}_{\mathrm{c}}$ is equivalent to $\mathcal{N}$ and hence the unique prime subnetwork of $\mathcal{N}$.*

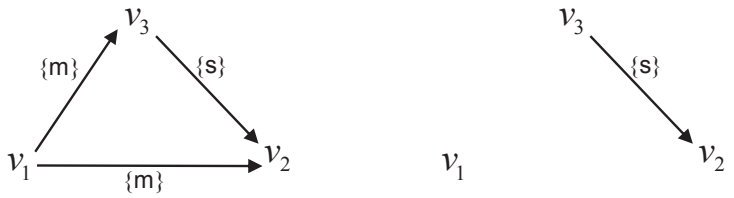
Proof. Suppose $c_1, c_2, \ldots, c_k$ are the redundant constraints of $\mathcal{N}$. Let $\mathcal{N}_0 = \mathcal{N}$ and $\mathcal{N}_{i+1} = \mathcal{N}_i \setminus \{c_{i+1}\}$ for $0 \leq i \leq k$. Note that $\mathcal{N}_k$ is precisely $\mathcal{N}_{\mathrm{c}}$, the core of $\mathcal{N}$. Assume on the contrary that $0 \leq i < k$ is the largest integer such that $\mathcal{N}_i$ is equivalent to $\mathcal{N}$.

Suppose $c_{i+1} = (xRy)$ and (xSy) is the corresponding constraint in $\mathcal{N}_{\mathrm{m}}$, the minimal network of $\mathcal{N}$. Note that c_{i+1} is also in $\mathcal{N}_i$. By Proposition 4.12 we

know (xSy) is redundant in $\mathcal{N}_{\mathrm{m}}$ since (xRy) is redundant in $\mathcal{N}$. Because $\mathcal{N}_{\mathrm{m}}$ is also the minimal network of $\mathcal{N}_i$, by Proposition 4.12 again we know (xRy) is redundant in $\mathcal{N}_i$. This means that $\mathcal{N}_{i+1}$ is equivalent to $\mathcal{N}_i$, hence $\mathcal{N}$. This contradicts our assumption that $i < k$ is the largest integer such that $\mathcal{N}_i$ is equivalent to $\mathcal{N}$. Therefore, $i = k$ and $\mathcal{N}_{\mathrm{c}}$ is equivalent to $\mathcal{N}$. □

As we have noted in the beginning of this section, for non-distributive subalgebras, the conclusion above is not always true, i.e. the core of a QCN might not be equivalent to it. Moreover, for IA, RA, and CRA, the above conclusion is not true even for QCNs with basic relations.

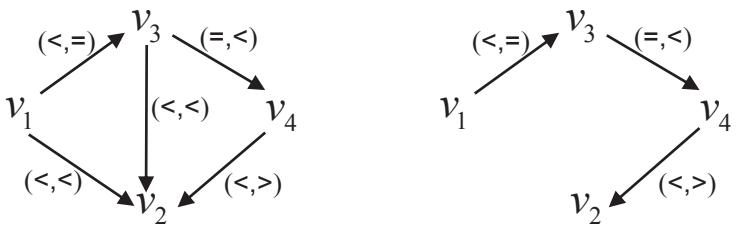
Example 4.5. Consider the IA QCN in Figure 4.6. Both the constraints $(v_1 \mathsf{m} v_2)$ and $(v_1 \mathsf{m} v_3)$ are redundant in the original QCN, because $\mathsf{m} \diamond \mathsf{s} = \mathsf{m}$ and $\mathsf{m} \diamond \mathsf{si} = \mathsf{m}$. However, the core shown in Figure 4.6(b) is obviously not equivalent to the original network. This is due to the fact that the redundancy of these two constraints is determined by each other. A similar example for QCN over RA can also be constructed by using the relations in the form $R \otimes \mathsf{id}_U$.



(a) The original IA QCN. (b) The core.

Figure 4.6: An example showing the core of a IA QCN over B_{IA} is not equivalent to the QCN.

Example 4.6. In the QCN in Figure 4.7(a), both $v_1(<,<)v_2$ and $v_3(<,<)v_2$ are redundant. However, the core shown in Figure 4.7(b) is not equivalent to the original QCN. We think that the circumstance comes from the partial violation of the all-different condition.



(a) The original IA QCN. (b) The core.

Figure 4.7: An example showing the core of a CRA QCN over $\mathsf{B}_{\mathrm{CRA}}$ is not equivalent to the QCN.

4.4 An Efficient Algorithm for Computing the Core

We have shown that the core of a QCN over a tractable subclass of PA or RCC5/8 can be found in $O(n^4)$ or $O(n^5)$ time. As an improvement, we propose an $O(n^3)$ time algorithm for the distributive subalgebras of PA or RCC5/8 to compute the core, shown as $\mathsf{Core}(\mathcal{N})$ in Algorithm 6. To identify redundant constraints in the input QCN $\mathcal{N}$, this algorithm first applies the result in Proposition 4.8, i.e. $(v_i S_{ij} v_j)$ is redundant in $\mathcal{N}_{\mathrm{p}}$ iff $S_{ij} = \bigcap\{S_{ik} \diamond S_{kj} : k \neq i, j\}$, to find redundant constraints in the a-closure of $\mathcal{N}$. Then by Proposition 4.11, the corresponding constraints in $\mathcal{N}$ is also redundant. The core is computed by directly removing all the redundant constraints. It can be easily seen that the algorithm runs in $O(n^3)$ time.

Later, Sioutis et al. [113] proposed a more efficient algorithm based on chordal graph and **PPC**. In particular, they have the following conclusion.

Proposition 4.14 ([113])**.** *Let $\mathcal{S}$ be a distributive subalgebra of any one of the calculi PA and RCC5/8. Suppose $\mathcal{N}$ is an all-different QCN over $\mathcal{S}$ and is* **PPC** *w.r.t. $G = (V, E)$, where G is a chordal graph s.t. $G_{\mathcal{N}} \subseteq G$. In addition, suppose $\mathcal{N}_p = \{(v_i S_{ij} v_j)\}$ is the a-closure of $\mathcal{N}$. Then, for $(v_i, v_j) \in E$, the constraint $(v_i S_{ij} v_j)$ is redundant in $\mathcal{N}_p$ iff $S_{ij} = \bigcap\{S_{ik} \diamond S_{kj} : k \neq i, j,\ \{v_i, v_k\}, \{v_k, v_j\} \in E\}$.*

Algorithm 6: Core($\mathcal{N}$), an algorithm for finding prime subnetworks.

Input: A consistent QCN $\mathcal{N} = \{v_i R_{ij} v_j : 1 \leq i, j \leq n\}$ with $V = \{v_i : 1 \leq i \leq n\}$ over a distributive subalgebra $\mathcal{S}$ of PA or RCC5/8.

Output: $\mathcal{N}_{\mathrm{c}}$: the core of $\mathcal{N}$.

```
Redun ← ∅;
N_p ← PC(N);
foreach constraint (v_i S_ij v_j) ∈ N_p do
    Q_ij ← ⋆;
    foreach variable v_k ∈ V \ {v_i, v_j} do
        Q_ij ← Q_ij ∩ S_ik ◇ S_kj;
        if Q_ij = S_ij then
            Redun ← Redun ∪ {(v_i R_ij v_j)};
            break the inner loop;
        end
    end
end
N_c ← N \ Redun.
```

To use this result to efficiently calculate the core, it is important to know how to efficiently check if $\mathcal{N}$ is an all-different QCN, i.e. $(\forall i, j)[(i \neq j) \rightarrow (\mathcal{N} \not\models (v_i \,\mathsf{id}_{\mathcal{U}}\, v_j))]$ (cf. (3.8)). Otherwise the above conclusion will not improve the efficiency of the algorithm Core for arbitrary QCNs. Fortunately, it suffices to only check whether there are pairs of v_i, v_j such that $(v_i \,\mathsf{id}_{\mathcal{U}}\, v_j)$ is in the $\mathbf{PPC}$ subnetwork.

Proposition 4.15. *Let $\mathcal{S}$ be a distributive subalgebra of PA or RCC5/8, and $\mathcal{N}$ is a consistent QCN over $\mathcal{S}$. Suppose that $G = (V, E)$ is a chordal graph s.t. $G(\mathcal{N}) \subseteq G$ and $\mathcal{N}_{\mathrm{p}}^G$ is the $\mathbf{PPC}$ subnetwork of $\mathcal{N}$ w.r.t. G. Then $\mathcal{N} \not\models (v_i \mathsf{id}_{\mathcal{U}} v_j)$ for any $v_i \neq v_j \in V$ iff $(u \mathsf{id}_{\mathcal{U}} v) \notin \mathcal{N}_{\mathrm{p}}^G$ for any $\{u, v\} \in E$.*

Proof. Note that, by Theorem 4.5, in $\mathcal{N}_{\mathrm{p}}^G$ the relations on the edges of the chordal graph are minimal labels. It is easy to see that if $\mathcal{N} \not\models (v_i \mathsf{id}_{\mathcal{U}} v_j)$ for any $v_i \neq v_j \in V$ then $(u \mathsf{id}_{\mathcal{U}} v) \notin \mathcal{N}_{\mathrm{p}}^G$ for any $\{u, v\} \in E$.

For the "if" direction, suppose on the contrary that $\mathcal{N} \models (v_i \mathsf{id}_{\mathcal{U}} v_j)$ for some $v_i \neq v_j \in V$. Then $\{v_i, v_j\} \notin E$, because $(v_i \mathsf{id}_{\mathcal{U}} v_j) \notin \mathcal{N}_{\mathrm{p}}^G$ for any $\{v_i, v_j\} \in E$. Without loss of generality, let us suppose $\{v_i, v_j\}$ is the first edge that was added into E and that generates new triangles, when we try to complete G and obtain a-closure according to the proof of Proposition 3.21. Then for any $k \neq i, j$ and $\{v_i, v_k\}, \{v_k, v_j\} \in E$, due to **PPC** of $\mathcal{N}_{\mathrm{p}}^G$ we have $\mathsf{id}_{\mathcal{U}} \diamond S_{ik} = S_{jk}$, where S_{ik} and S_{jk} are the relations in $\mathcal{N}_{\mathrm{p}}^G$. This will result in that $S_{ik} \diamond S_{kj} = S_{ik} \diamond S_{ik}^{-1}$, which contains **O** for RCC5 or RCC8, and contains $\star$ for PA by Lemma 3.14. This means that $\bigcap\{S_{ik} \diamond S_{kj} : k \neq i, j \text{ and } \{v_i, v_k\}, \{v_k, v_j\} \in E\} \neq \mathsf{id}_{\mathcal{U}}$. This is a contradiction to the construction of the relation between v_i and v_j. □

Therefore, we can determine if a QCN is all-different by using its **PPC** subnetwork, which will be more efficient than using the **PC** subnetwork.

Note that if $\{v_i, v_j\} \notin E$, then $(v_i S_{ij} v_j)$ is always redundant in $\mathcal{N}_{\mathrm{p}}$. Regarding the core of a QCN, Sioutis et al. [113] have discussed the following result and a more efficient algorithm that makes use of the graph structure of QCNs to find the core.

Theorem 4.16 ([113]). *Let $\mathcal{S}$ be a distributive subalgebra of PA or RCC5/8. Suppose $\mathcal{N}$ is an all-different QCN over $\mathcal{S}$ and $G = (V, E)$ a chordal graph s.t. $G(\mathcal{N}) \subseteq G$. Assume that $\mathcal{N}_p^G$ is the **PPC** subnetwork of $\mathcal{N}$ w.r.t. G and $\mathcal{N}_{\mathrm{c}}$ is the core of $\mathcal{N}$. Then a constraint $(v_i R_{ij} v_j)$ in $\mathcal{N}$ is also in $\mathcal{N}_{\mathrm{c}}$ iff $\{v_i, v_j\} \in E$ and $S_{ij} \neq \bigcap\{S_{ik} \diamond S_{kj} : k \neq i, j,\ \{v_i, v_k\}, \{v_k, v_j\} \in E\}$, where $(v_i S_{ij} v_j)$, $(v_i S_{ik} v_k)$, and $(v_k S_{kj} v_j)$ are the constraints in $\mathcal{N}_p^G$.*

4.5 Summary and Discussion

In this chapter, we discussed the redundancy problem of distributive subalgebras. For any all-different QCN over one of the distributive subalgebras of

PA and RCC5/8, we showed that there exists a unique prime subnetwork of the QCN that is exactly the core of it. On the other hand, for non-distributive subalgebras and distributive subalgebras of IA, RA, and CRA, we have provided examples to show the conclusion generally does not hold. We have also devised an $O(n^3)$ time algorithm to construct the unique prime subnetwork (i.e. the core) of any QCN over one of the distributive subalgebras of PA and RCC5/8.

In [77] and [113], the effectiveness and the efficiency of the algorithm Core and the one using the $\mathbf{PPC}$ subnetwork to construct the core of a QCN have been thoroughly verified on several real-world datasets. In particular, in [77], the evaluation was conducted on the complete basic RCC8 QCNs of two kinds of datasets, i.e. the structured dataset about statistical areas that contains no overlapping regions and the unstructured footprint dataset derived from social media, which contains a variety of regions of different size and shapes but sharing almost no adjacent boundaries. It was observed that on both datasets there are more than 90% redundant constraints. There is also a correlation between the number of **PO** relations and the number of redundant constraints on the dataset of footprint, i.e. the less **PO** constraints the more redundant relations there are. This can be explained by that the weak compositions of **PO** with other relations are usually not very restrictive and always contains **PO**. Moreover, it is easy to see from the composition table of RCC8 that, if a QCN contains only **DC**, **EC**, **PO** relations, then there will be no redundant constraints at all. In [113], the algorithms are tested on larger and sparser QCNs that also contain non-basic RCC8 relations. They observed a speed-up of the algorithm using the $\mathbf{PPC}$ subnetwork over the algorithm Core. Interestingly, although the density of the QCNs is already very low, there are still many redundant constraints in some of these QCNs, with a highest reduction rate of 65.66%.

With a high reduction rate of constraints on real-world datasets, prime subnetworks can be used to simplify the structure of QCNs so that other tasks whose performance is strongly related to the structure, e.g. the topological adjustment of spatial scenes and comparison of different QCNs, as we have mentioned in Section 2.5.4. This will make the comprehensive knowledge base be able to better support other applications.

However, removing redundant constraints from QCN will make the retrieval of the removed constraints inefficient. In fact, to restore a removed constraint from the prime subnetwork, we need to enforce either DPC or PPC on the QCN, which has a worst case performance $O(n^3)$. When the number of variables becomes larger, the retrieval of the removed constraints will quickly become inefficient. In fact, in [77], on the footprint dataset where the regions contains less vertices, the retrieval of the relations from the prime subnetwork becomes less efficient than direct geometric computation. On several small datasets of 600 variables, which will be used in the following chapters, we have also observed that retrieving relations with the prime subnetworks takes more than 10^7ns, which is longer than most of the computation time from geometric representation. In other words, the prime subnetwork representation is not suitable to be used as a representation that support efficient retrieval of relations. Moreover, it is still an open question if the prime subnetwork representation is applicable for CDC QCNs, which are very common in real-world applications to describe directional information.

Therefore, we need other techniques to support efficient relation retrieval for any two given variables or entities, while keeping the storage size in a relatively low level. In the next two chapters, we will discuss two promising candidates that can reduce the storage space as well as support efficient relation retrieval.

Chapter 5

Compact Representation: Encoding with MBRs

5.1 Introduction

In the literature, we have seen that the vast majority of the researches on QSTR have been devoted to the efficiency of the reasoning time. These researches presume that the qualitative relations are always explicitly stored in QCNs, and many previous applications of QSTR only deal with a relatively small number of variables [48, 102, 121, 127] or on sparse structured QCNs [111–113]. On the other hand, although QCNs derived from sources like text descriptions [54, 62, 70] usually have a sparse structure, there are still cases that the QCNs are complete and have a large number of variables and relations in many real-world applications. The large size of the QCNs severely restricts the ability of knowledge bases or applications to handle qualitative spatial information. Specifically, for the fundamental task of retrieving the relation between two variables, the ideal solution is to have an explicit relation between any two variables. However, because the size of a complete QCN is quadratic if the

relations are binary and even cubic if the relations are ternary, the complete QCN representation will cost a lot of storage space for spatial or temporal scenes with hundreds of thousands of variables. The QCNs would be too costly or infeasible to fit into fast accessible storage, which in turn makes the retrieval of the relations much less efficient.

Note that although the prime subnetwork discussed in the previous chapter can simplify the QCN, it is not suitable for the task of retrieving the relations, because to restore the removed relations this approach needs to run $O(n^3)$ time algorithms, which will soon become infeasible when the number of variables n becomes large. On the other hand, when the geometric representation of spatial entities is available, the current applications such as GIS [104] rely on a compromise that uses geometric algorithms to directly compute the relations. However, the online computation could be expensive, because it requires computation time that is proportional to the number of vertices in the polygon, which can be quite large (e.g. in our test data we have encountered polygons with more than $30{,}000$ vertices).

Therefore, an important challenge for QSTR is to develop methods for representing qualitative spatial information more compactly while the relations can still be retrieved efficiently, i.e. to solve the compact representation problem specified in Chapter 2.

In this chapter, we present a simple but powerful technique to represent both topological and directional information compactly, while supporting efficient retrieval of the relations between spatial objects. The technique focused on dealing with QCNs derived from geometric information. In this case, qualitative relations mainly serve as a layer for query answering and human-friendly interaction (e.g., [16]), and for respecting privacy concerns in cases such as the precise locations or boundaries are sensitive. It is worth noting that, unlike

previous chapters, our scope in this chapter is restricted to basic relations in RCC8 or CDC, and where exact regions are given.

This chapter is organised as follows. In Section 5.2 we briefly review the state-of-the-art approach for the compact representation problem, i.e. the spatial clustering index proposed by Fogliaroni [43]. Then we present our approach in Section 5.3 and analyse its effectiveness in theory and by experiments. In Section 5.4, we discuss how to efficiently retrieve the relation between two entities with the compact representation and evaluate it on real-world datasets. In Section 5.5, we propose several derivatives of our approach. Section 5.6 summarises this chapter and discusses the problem of the current methods.

Bibliographic Note. The work reported in this chapter is based on the joint work with Matt Duckham, Sanjiang Li, and Steven Schockaert, which was first presented in the following publication.

- Zhiguo Long, Matt Duckham, Sanjiang Li, and Steven Schockaert: Indexing Large Geographic Datasets with Compact Qualitative Representation, International Journal of Geographical Information Science, 2016, vol. 30, no. 6, pp. 1072–1094.

Sanjiang Li and I initiated the research in the paper about compact representation. Guided by Sanjiang Li, I proposed the core idea and devised the algorithm in the paper. Matt Duckham provided constructive advice and helped on the empirical evaluation, which was conducted by myself. Steven Schockaert also provided valuable advice on the empirical evaluation and improvements of the techniques.

5.2 Spatial Clustering Index

Before discussing our approach, we first have a brief review of the state-of-the-art approaches for this problem in the literature. Fogliaroni et al. [43, 44] proposed the *spatial clustering index* to provide a more compact qualitative representation. The approach uses the so-called *clustering relations* to reduce the storage of qualitative relations between regions. Given a qualitative calculus $\mathcal{M}$, a relation $\alpha \in \mathcal{M}$ is a clustering relation if it is downward closed under set inclusion, i.e. for any regions a and b, from $(a, b) \in \alpha$ and $a' \subseteq a, b' \subseteq b$, we have that $(a', b') \in \alpha$. It is straightforward to show that RCC8 only has one clustering relation, i.e. **DC**, and CDC has exactly four, i.e. the single tile relations NW, NE, SW, and SE.

Given a spatial dataset containing the set of regions $\mathcal{D} = \{o_i : i \in I\}$, the spatial clustering index builds a spatial clustering structure on $\mathcal{D}$ and then selectively computes and stores the relations between the regions. It makes use of some auxiliary geometric shapes, called "index tiles" to help detect the clustering relations between the regions that are *associated with* the index tiles. Regions are associated with these index tiles according to a certain predefined strategy, such as the strategy of considering the intersection between a region and the index tile. In general, spatial clustering index is constructed as follows for RCC8 or CDC.

1. Build a spatial clustering structure $\mathcal{I} = \{(t_j, C_j) : j \in J\}$, where each t_j is an *index tile* with $j \in J$ (J is a set of integers for numbering t_j) and $C_j \subseteq \mathcal{D}$ a cluster of regions associated with t_j according to some strategy, and $\mathcal{D} = \{o_i : i \in I\}$ is covered by $\{t_j : j \in J\}$, i.e. $\bigcup_{i \in I} o_i \subseteq \bigcup_{j \in J} t_j$.

2. For each pair $(i, j) \in J \times J$, execute the following steps:

   (a) If $(t_i, C_i) = (t_j, C_j)$, then compute and store the RCC8/CDC relation

between every two regions in $C_i (= C_j)$;

(b) If (t_i, C_i) and (t_j, C_j) are different, then compute the RCC8/CDC relation between t_i and t_j;

i. If the relation between t_i and t_j is a clustering relation, then store this relation[1] for t_i and t_j, and continue for the next pair of i, j;

ii. If the relation between t_i and t_j is not a clustering relation, then compute and store the relations between every region in C_i and every region in C_j.

As we can see from the above steps in 2(b), the relations between some regions will not need to be computed and stored if the index tiles for these regions are in a clustering relation. Then the storage space to represent the RCC8/CDC relations will be reduced. Different types of index tiles and clustering structures will affect the reduction rate. Fogliaroni [43] implemented two instances of spatial clustering index: one is grid-based and the other is *R*-tree* based. As R*-trees [9] are only one of the variants of R-trees, the R*-tree clustering index can be straightforwardly extended to methods based on other variants of R-trees [61]. In this thesis, we will refer to this class of methods as the *R-tree clustering index*.

The *grid clustering index* partitions the plane into grid cells of equal size, and uses the grid cells as index tiles (see Figure 5.1). To build a spatial clustering structure, different strategies can be exploited. For RCC8, the grid clustering index uses the strategy that a region r_i is associated with an index tile t iff $t \cap \mathsf{mbr}(r_i) \neq \varnothing$; for CDC, it uses the strategy that a region r_i is associated with

[1] It is optional whether to store the clustering relation between tiles. Based on different query strategies, we may either not need this piece of information such as for the strategy later used in our experiments, or need it for some other strategies like the one in [43].

an index tile t iff $\mathsf{mbr}(r_i)$ and t have a common interior point. For both RCC8 and CDC,

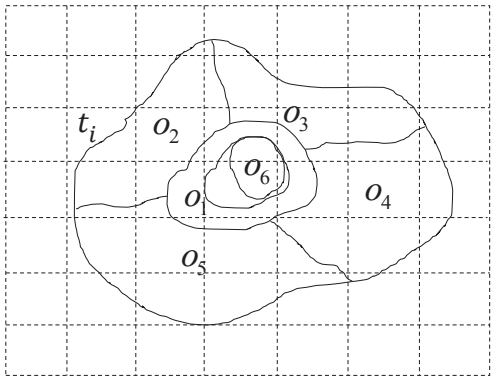
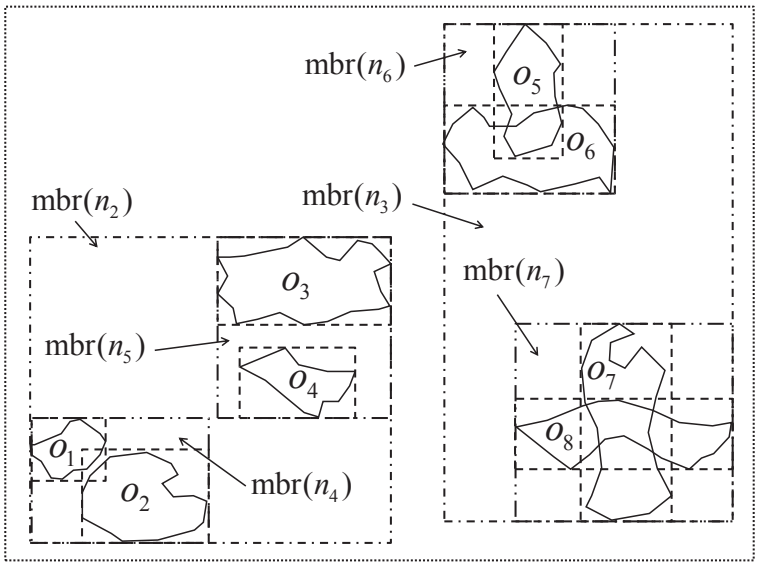
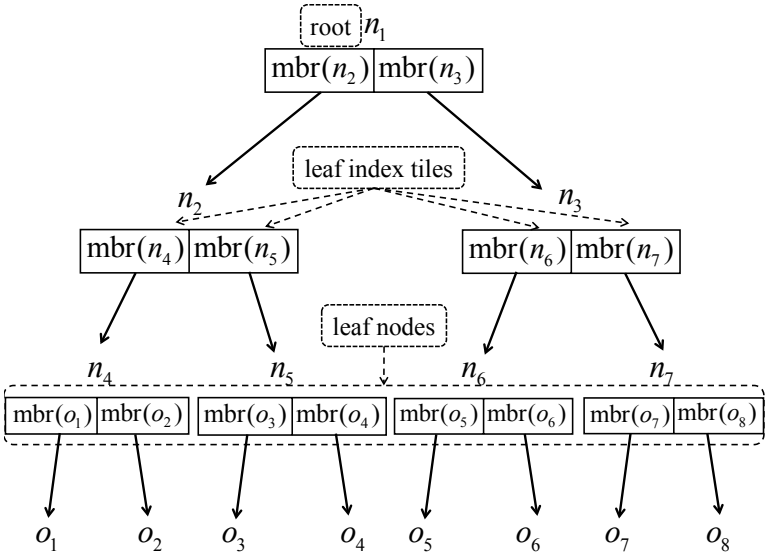


Figure 5.1: Illustration of grids in the plane and grid index tiles t_i.

The R-tree clustering index uses an instance of the R-tree family to hierarchically construct index tiles. To understand how to build the R-tree clustering index, we first take a look at the structure of the original R-tree in the two-dimensional plane. For example, Figure 5.2(a) shows a set of regions $\{o_1, \ldots, o_8\}$ in the plane, and Figure 5.2(b) gives an R-tree structure for these regions, where the MBRs have been given in Figure 5.2(a). As illustrated in the figure, an R-tree is a balanced tree structure with root, non-leaf nodes, and leaf nodes that are on the same level. Moreover, each leaf node of an R-tree contains an array of leaf entries in the form of $(\mathsf{mbr}(a), \mathrm{oid}(a))$, where a is an object and $\mathrm{oid}(a)$ is the address of a. For example, nodes n_4, n_5, n_6, n_7 in Figure 5.2(b) are leaf nodes, and $o_1, \ldots, o_8$ are objects. Each non-leaf node contains an array of entries in the form of $(\mathsf{mbr}(n_i), \mathrm{nodeid}(n_i))$, where n_i is a child node of the current one, $\mathrm{nodeid}(n_i)$ is the address of n_i, and $\mathsf{mbr}(n_i)$ is the MBR of the MBRs in the entry array of n_i. The R-tree clustering index then uses the MBRs in an R-tree of the regions as the index tile, e.g. $\mathsf{mbr}(n_2), \ldots, \mathsf{mbr}(n_7)$ in Figure 5.2(b). R*-tree has the same structure specification as the original R-tree, which applies a different approach to split a node with too many entries.



(a) Regions and tiles in the plane.



(b) The corresponding R-tree.

Figure 5.2: Illustration of an R-tree.

To build a spatial clustering structure, for both RCC8 and CDC, the R-tree clustering index hierarchically builds the clustering index and has two strategies. A *leaf (index) tile* in the R-tree clustering index is $\mathsf{mbr}(n_i)$, where n_i is a parent node of a leaf node, e.g. in Figure 5.2 $\mathsf{mbr}(n_i)$ for $i = 4, 5, 6, 7$ are leaf index tiles. For leaf index tiles, in the first strategy, a region r_i is associated with a leaf index tile t only if $\mathsf{mbr}(r_i)$ and t have a common interior point; in the second strategy, a region r_i is associated with a leaf index tile t only if $\mathsf{mbr}(r_i)$ is contained in t. For a non-leaf index tile t, in both strategies of the R-tree clustering index, a lower level index tile t' is associated with t only if t' is contained in t. The strategies are summarised in Table 5.1.

Table 5.1: Strategies to associate r_i (or t') with t, where t and t' are index tiles, t° is the interior of t, and r_i is a region.

Strategies	RCC8	CDC
grid clustering index	$t \cap \mathsf{mbr}(r_i) \neq \varnothing$	$t^\circ \cap \mathsf{mbr}(r_i) \neq \varnothing$
leaf tile for R-tree clustering index	$t^\circ \cap \mathsf{mbr}(r_i) \neq \varnothing$	
	$\mathsf{mbr}(r_i) \subseteq t$	
non-leaf tile for R-tree clustering index	$t' \subseteq t$	

As has been observed in [43], the performance of the spatial clustering index (i.e. the reduction ratio of the storage space) strongly depends on the quality of the clustering structure. A bad clustering structure will not only result in many repeated considerations of region pairs, but also fail to associate a sufficient number of regions to index tiles that are in clustering relations. It was conjectured in [43] and confirmed by some experimental results that the grid clustering index has the best performance when the size of index tiles is about the average size of the regions in the dataset.

There is another variant of the spatial clustering index for point objects developed by Al-Salman [1], which aims to obtain a better cluster by using a more sophisticated clustering strategy (i.e. the density-based approach DB-

SCAN [41]) and using the concave hull [30, 36] of each cluster as the index tile rather than using the MBR for RCC8 case. However, the effectiveness of this approach for regions remains a question and hence we will not discuss it here.

5.3 The MBR-Based Approach

In the following, we present another alternative, called the MBR-based approach, for solving the compact representation problem.

The *MBR-based approach* (denoted by MA) was inspired by the following observations:

1. MBRs can usually be obtained in real-world applications at very low cost, and can be stored linearly with respect to the number of involved objects.

2. While the number of region pairs whose MBRs do not have common point (or interior point) depends on the nature of the considered datasets, we found that many real-world datasets contain a large number of such pairs (e.g. administrative areas), and hence the MBR of a region usually only intersects with a small number of the MBRs of other regions.

3. The RCC8 relation between two regions can be unambiguously inferred from the RCC8 relation between their MBRs if the two MBRs have no common point.

4. The CDC relation between two regions can be unambiguously inferred from the CDC relation between their MBRs (and sometimes the MBRs of their connected components) if the two MBRs have no common interior point.

5. Calculating the RCC8/CDC relation between two MBRs is much easier and more efficient than calculating the RCC8/CDC relation between two arbitrary regions.

Algorithm 7 shows the main steps of MA. In Line 3, we first find all pairs (o_i, o_j) such that the corresponding MBRs have a common point, and then calculate the RCC8 relations between such pairs in Lines 4-6. Similarly, we calculate the CDC relations in Lines 9-12. Note that since CDC relations are not closed under converse [86], we need to calculate and store both the CDC relation from o_i to o_j and that from o_j to o_i.

Algorithm 7: MA($\mathcal{D}$), an algorithm for the MBR-based approach to construct compact representation.

Input: A set of regions $\mathcal{D} = \{o_1, \ldots, o_n\}$.
Output: A compact representation, from which RCC8/CDC relations can be derived.

```
1  Obtain the MBRs {mbr(o_1), ..., mbr(o_n)} of D = {o_1, ..., o_n};
2  case RCC8 do
3  |  Pair ← {(o_i, o_j) : i < j, mbr(o_i) ∩ mbr(o_j) ≠ ∅};
4  |  foreach (o_i, o_j) ∈ Pair do
5  |  |  Calculate and store the RCC8 relation between o_i and o_j;
6  |  end
7  end
8  case CDC do
9  |  PairInt ← {(o_i, o_j) : i < j, (mbr(o_i))° ∩ (mbr(o_j))° ≠ ∅};
10 |  foreach (o_i, o_j) ∈ PairInt do
11 |  |  Calculate and store the CDC relation from o_i to o_j and that from
   |  |  o_j to o_i;
12 |  end
13 end
```

The idea of the algorithm is related to the widely used MBR pre-processing technique, which is applied in GIS systems to filter out candidate answers for queries. However, while the standard pre-processing method aims to improve the computation time of geometric algorithms, we propose to use MBRs to

construct a more compact representation of qualitative spatial information. We will show that it can significantly reduce the storage and outperform the spatial clustering index approach by Fogliaroni et al. [43], both in theory and in experiments. We will also illustrate the efficiency of retrieving the relation between any two given variables by using the resulting compact representation.

5.3.1 Correctness

We need to show that the algorithm is correct. That is, after applying the algorithm to a set of regions $\mathcal{D} = \{o_1, ..., o_n\}$, the RCC8/CDC relation between any two regions o_i and o_j can be correctly retrieved from the resulting representation. In other words, the relation is either stored explicitly or can be unambiguously inferred from the corresponding MBRs $\mathsf{mbr}(o_i)$ and $\mathsf{mbr}(o_j)$ (and sometimes the MBRs of the connected components of the regions). In the following, we show that the relations that are not stored can be inferred from the relations between MBRs.

It is easy to see that, as the following proposition shows, for RCC8, the non-stored topological relations can always be inferred from the topological relations between the corresponding MBRs.

Proposition 5.1 ([76, 95])**.** *Given two (connected or disconnected) regions a and b, if $\mathsf{mbr}(a)$ **DC** $\mathsf{mbr}(b)$, i.e. $\mathsf{mbr}(a) \cap \mathsf{mbr}(b) = \varnothing$, then a**DC**b.*

The following proposition shows that for CDC and connected regions, a similar conclusion also holds. In the following, $(\mathsf{mbr}(a))^\circ$ denotes the interior of the MBR $\mathsf{mbr}(a)$.

Proposition 5.2. *Given two regions a and b, if a is connected and $(\mathsf{mbr}(a))^\circ \cap (\mathsf{mbr}(b))^\circ = \varnothing$, then $\delta(\mathsf{mbr}(a), \mathsf{mbr}(b)) = \delta(a, b)$, i.e. the CDC relation from a to b*

is the same as that of $\mathsf{mbr}(a)$ *to* $\mathsf{mbr}(b)$.

Proof. It is easy to see that $\delta(a,b) \subseteq \delta(\mathsf{mbr}(a), \mathsf{mbr}(b))$. In the following, we show $\delta(\mathsf{mbr}(a), \mathsf{mbr}(b)) \subseteq \delta(a,b)$.

Because $(\mathsf{mbr}(a))^\circ \cap (\mathsf{mbr}(b))^\circ = \varnothing$, $\mathsf{mbr}(a)$ must be contained in one of the following tile regions: $t_{\mathsf{NW}} \cup t_{\mathsf{N}} \cup t_{\mathsf{NE}}$, $t_{\mathsf{NE}} \cup t_{\mathsf{E}} \cup t_{\mathsf{SE}}$, $t_{\mathsf{SE}} \cup t_{\mathsf{S}} \cup t_{\mathsf{SW}}$, $t_{\mathsf{NW}} \cup t_{\mathsf{W}} \cup t_{\mathsf{SW}}$.

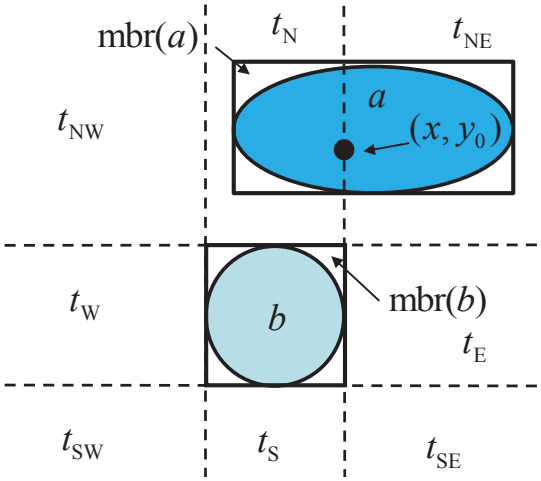


Figure 5.3: Illustration of the proof of Proposition 5.2.

Without loss of generality, suppose $\mathsf{mbr}(a)$ is contained in $t_{\mathsf{NW}} \cup t_{\mathsf{N}} \cup t_{\mathsf{NE}}$ (see Figure 5.3 for illustration). Let t_i be one of these three tiles. There are two cases when $(\mathsf{mbr}(a))^\circ \cap t_i^\circ \neq \varnothing$.

Case 1. $\mathsf{mbr}(a) \subseteq t_i$. Then we have $a^\circ \cap t_i^\circ \neq \varnothing$ and thus the tile name of t_i is also in $\delta(a,b)$.

Case 2. $\mathsf{mbr}(a) \cap t_i \neq \varnothing$ and $\mathsf{mbr}(a) \not\subseteq t_i$. Let y^- and y^+ be the minimum and maximum y-coordinates of $\mathsf{mbr}(a)$ respectively. Then $\exists x$ s.t. the points (x,y) with $y \in (y^-, y^+)$ satisfy $(x,y) \in t_i^\circ$ and $(x,y) \in (\mathsf{mbr}(a))^\circ$. Because a is connected, we know that there is some $y_0 \in (y^-, y^+)$ s.t. $(x, y_0) \in a^\circ$. This means $a^\circ \cap t_i^\circ \neq \varnothing$.

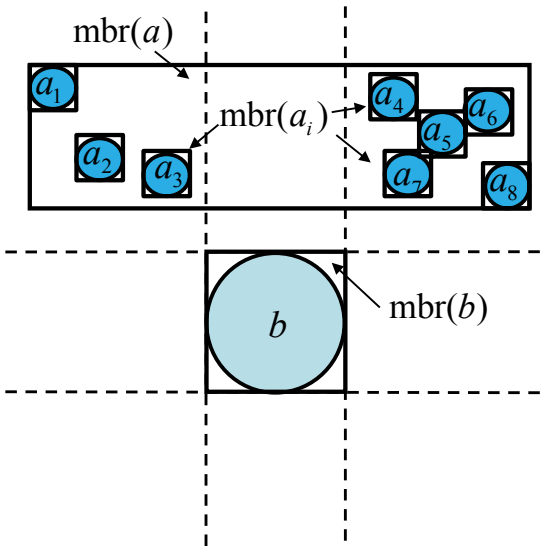


Figure 5.4: The CDC relation from $\mathsf{mbr}(a)$ to $\mathsf{mbr}(b)$ is $\{\mathsf{NW}, \mathsf{N}, \mathsf{NE}\}$, while the CDC relation from a to b is $\{\mathsf{NW}, \mathsf{NE}\}$.

Therefore, we have shown that $\delta(\mathsf{mbr}(a), \mathsf{mbr}(b)) \subseteq \delta(a, b)$. □

For possibly disconnected regions, however, we cannot always get the correct CDC relation merely from the MBRs of the regions. Figure 5.4 shows such an example, where the CDC relation from a to b is $\{\mathsf{NW}, \mathsf{NE}\}$ but the CDC relation from $\mathsf{mbr}(a)$ to $\mathsf{mbr}(b)$ is $\{\mathsf{NW}, \mathsf{N}, \mathsf{NE}\}$. In this case, we need to take the connected components of a into consideration.

Lemma 5.3. *Given two possibly disconnected regions a, b, suppose $a_1, \ldots, a_k$ are the connected components of a. Then $\delta(a, b) = \bigcup_{i=1,\ldots,k} \delta(a_i, b)$, where each $\delta(a_i, b)$ is a subset of $\{NW, N, NE, W, O, E, SW, S, SE\}$.*

Proof. Because $a^\circ \cap t_i^\circ \neq \varnothing$ if and only if there exists a connected component a_i of b s.t. $a_i^\circ \cap t_i^\circ \neq \varnothing$ (see, e.g., Figure 5.4), the conclusion is easy to see. □

The above lemma states that the CDC relation of disconnected regions can be obtained from the CDC relations of their connected components. Note that in the algorithm MA, we only need to store the CDC relation when $\mathsf{mbr}(a)$

and $\mathsf{mbr}(b)$ have no common interior point. In this case, $\mathsf{mbr}(a_i)$ and $\mathsf{mbr}(b)$ also do not have common interior point. Then by Proposition 5.2, $\delta(a_i, b) = \delta(\mathsf{mbr}(a_i), \mathsf{mbr}(b))$ and hence $\delta(a, b) = \bigcup_{i=1,\ldots,k} \delta(\mathsf{mbr}(a_i), \mathsf{mbr}(b))$ (see, e.g., Figure 5.4). In summary, for possibly disconnected regions, we have the following conclusion that the CDC relations not stored can always be inferred from the relations between the MBRs of the connected components of the regions.

Proposition 5.4. *Given two possibly disconnected regions a, b, suppose $a_1, \ldots, a_k$ are the connected components of a, and $\mathsf{mbr}(a)$ and $\mathsf{mbr}(b)$ have no common interior point. Then $\delta(a, b) = \bigcup_{i=1,\ldots,k} \delta(\mathsf{mbr}(a_i), \mathsf{mbr}(b))$.*

5.3.2 Effectiveness

Regarding the effectiveness of the algorithm to construct a compact representation of RCC8 and CDC relations, the major concern is the number of relations that need to be stored for a given dataset $\mathcal{D}$, since the geometric information and the MBRs scale linearly with the number of spatial entities. For convenience, we call such a number the *qualified size* of $\mathcal{D}$ for the corresponding algorithm (e.g., the complete QCN, MA or the grid/R-tree clustering indexes), and use it as the measure for the performance of the algorithms.

For the grid and R-tree clustering indexes, the qualified size is related not only to the characteristics of the dataset but also to the parameters chosen by the algorithms, such as the index tile size for the grid clustering structure and the maximal number of children allowed in a node of an R-tree. For MA, however, the qualified size is only related to a particular characteristic of the spatial geometric dataset, i.e. the *average intersection degree* as defined below.

Definition 5.5. Given a spatial dataset of n regions $\mathcal{D} = \{o_1, \ldots, o_n\}$, let $P = \{(\mathsf{mbr}(o_i), \mathsf{mbr}(o_j)) : \mathsf{mbr}(o_i) \cap \mathsf{mbr}(o_j) \neq \varnothing\}$ and $Q = \{(\mathsf{mbr}(o_i), \mathsf{mbr}(o_j)) :$

$(\mathsf{mbr}(o_i))^\circ \cap (\mathsf{mbr}(o_j))^\circ \neq \varnothing\}$. We call $|P|/n$ the average intersection degree of $\mathcal{D}$ for RCC8, and call $|Q|/n$ the average intersection degree of $\mathcal{D}$ for CDC.

Intuitively, the average intersection degree characterises that, in the dataset, the MBR of a region on average intersects with how many other MBRs. If the context of the specific calculus is clear, we will simply refer to $\bar{d} = |P|/n$ or $\bar{d} = |Q|/n$ as the average intersection degree of $\mathcal{D}$. The average intersection degree of $\mathcal{D}$ actually determines the qualified size of $\mathcal{D}$ for MA.

Proposition 5.6. *Given a spatial dataset of n regions $\mathcal{D}$, the qualified size of $\mathcal{D}$ for* MA *is $n\bar{d}/2$ for RCC8 relations and $n\bar{d}$ for CDC relations.*

Interestingly, for many real-world datasets, most regions only have a relatively small number of "neighbouring" regions and the average intersection degree tends to be quite small when compared to the number of regions in the configuration. For example, Figure 5.5 shows the distribution of the number of intersecting MBRs for a given MBR (i.e. the intersection degree of an MBR) for the administrative areas of Australia. We have $\bar{d} \approx 7.12$, which is quite small compared to the number of regions $n = 1{,}395$. As a result, the qualified size of this dataset for MA will be much smaller than the qualified size for the complete QCN representation.

Moreover, the qualified size of a given set of regions for MA is never larger than the qualified size for the grid or the R-tree clustering indexes, for both RCC8 and CDC.

Proposition 5.7. *Given a set of possibly disconnected regions $\mathcal{D} = \{o_1, ..., o_n\}$, then, for RCC8, the qualified size of $\mathcal{D}$ for* MA *is not larger than the qualified sizes of $\mathcal{D}$ for either the grid clustering index or the R-tree clustering index.*

Proof. We defer the proof to Appendix B.3. □

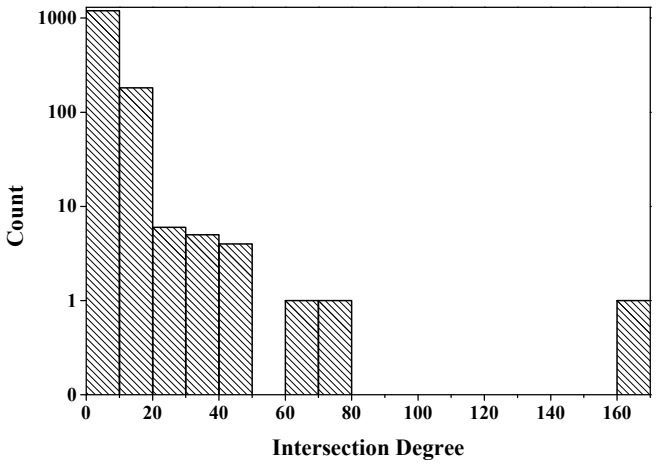


Figure 5.5: Distribution of intersection degree of the administrative areas of Australia.

A similar result also applies to CDC.

Proposition 5.8. *Given a set of possibly disconnected regions $\mathcal{D} = \{o_1, ..., o_n\}$, for CDC the qualified size of $\mathcal{D}$ for* MA *is not larger than the qualified sizes for either the grid or the R-tree clustering indexes.*

Proof. We also defer the proof to Appendix B.3. □

Finally, we note that the task of identifying all (interiorly) intersecting MBRs can be efficiently accomplished. In fact, this task is a classic problem known as the Rectangle Intersection Problem (or Rectangle Spatial Join Problem). There are several well-known efficient algorithms. Using the so-called interval trees or segment trees or priority search trees, we can identify all intersecting pairs of MBRs in time $O(n \log n + k)$, where k is the number of intersecting pairs. For example, see works in [10, 59, 60, 104]. Since on average there are only a small number of MBRs intersecting with a given MBR, the time needed is usually dominated by $O(n \log n)$. Moreover, for the many datasets, using brute-force search to find intersecting pairs of MBRs is already very efficient.

5.3.3 Empirical Evaluation

We compare the performance of MA with the grid and the R-tree clustering indexes proposed in [43], and also with the complete QCN representation.

When applying the grid clustering index, as in Chapter 6 of [43] we select a grid $\mathcal{G}$ that covers the dataset such that the size of each index tile in the grid is about the same as the average size of the regions in the dataset $\mathcal{D} = \{o_1, ..., o_n\}$. Following [43] and as specified in Table 5.1, the cluster of regions associated with an index tile t is exactly the set of regions $\{o_i\}$ such that $t \cap \mathsf{mbr}(o_i) \neq \varnothing$.

For the R-tree clustering index, since there are no generally accepted optimal parameter settings or tree building strategies, we use the efficient bulk-loading variant of R-tree, the STR R-tree [74], as an illustration of the performance of the R-tree clustering index. The STR R-tree is designed for static or a priori available objects, which is the case here, and it can efficiently build an R-tree such that only a small number of MBRs overlap. The implementation of STR R-tree used in the experiment is from JTS[2]. We use the default parameter setting of the implementation. Also, we build the R-tree clustering index structure from the root level of the tree, because, as suggested in [43], the algorithm will reduce more of the qualified size if it starts from a shallower level of the tree. Similar to the grid clustering index, the R-tree clustering index associates regions with index tiles by using the strategies specified in Table 5.1.

As noticed in [43], the grid clustering index has another weak point. In the grid clustering procedure, some pairs of regions may be simultaneously associated with several index tiles. This will result in these pairs of regions being repeatedly used for computation. If the number of such pairs is large, then the number of qualified relations would be so large that using the grid clustering index even becomes more expensive than the complete representation. In our

[2] http://www.vividsolutions.com/jts/JTSHome.htm

experiments, before calculating the spatial relation, we check whether a pair of regions has already been considered.

Datasets

We select four classes of real-world datasets for the evaluation. The first class (called Real-1) contains datasets that have relatively small average intersection degrees and various number of regions. The second class (called Real-2) contains datasets that have relatively large average intersection degrees and similar number of regions. The third class (called Real-3) contains datasets that have large number of regions and small average intersection degrees. The fourth class (called Real-4) contains datasets of larger number of regions than the ones in Real-2, and have relative large average intersection degrees.

Real-1 comprises administrative datasets of various sizes extracted from Global Administrative Areas (GADM[3]): Germany-adm3 (434 regions), Ukraine-adm2 (629 regions), Australia-adm2 (1395 regions), China-adm3 (2411 regions), and USA-adm2 (3145 regions). The average intersection degrees of all the five datasets are around seven. Real-2 consists of five datasets about environmental habitats extracted from the European Environment Agency (EEA[4]), containing approximately the same number of regions but with different average intersection degrees. In particular, the five datasets of Real-2 were selected to ensure a range of average intersection degrees. Each dataset contains around 600 regions, with average intersection degrees of respectively about 45, 106, 122, 180, and 205. Real-3 contains four datasets that have a large number of regions : Statistical Areas Level 1 of Australia (SA1; $54{,}772$ regions), New South Wales Mesh Blocks of Australia (NSWMB; $107{,}317$ regions), County Sub-

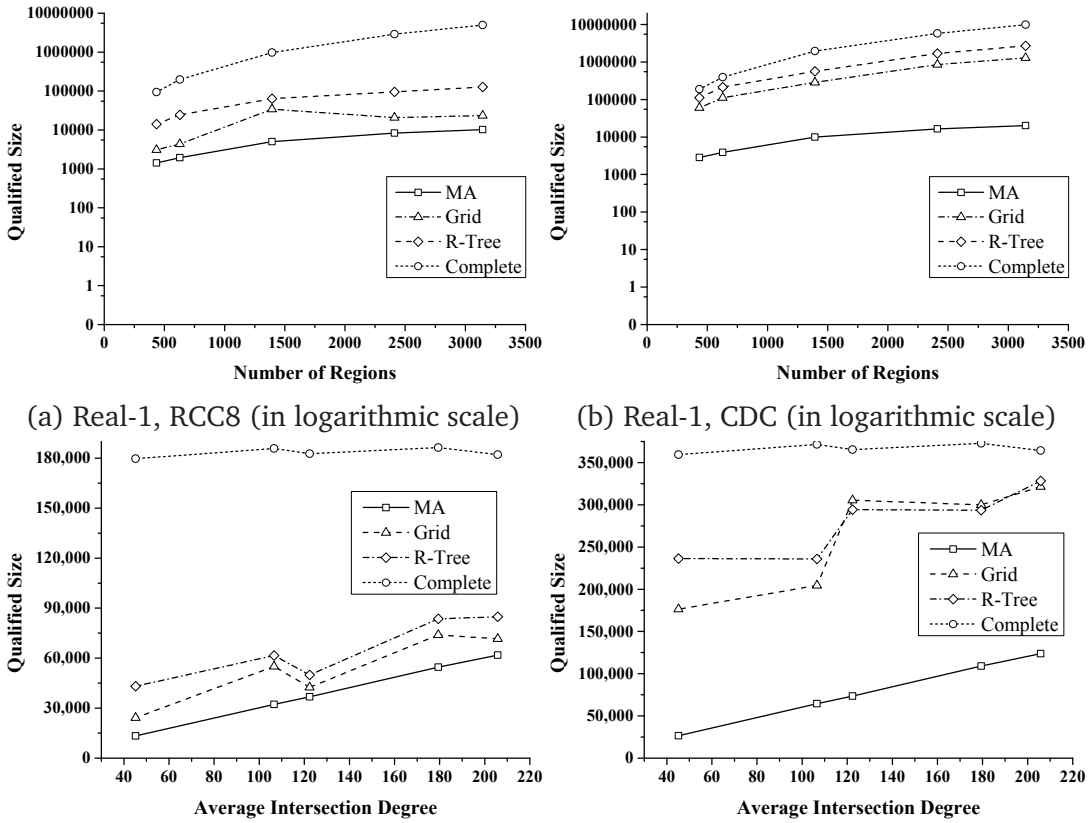
[3] http://www.gadm.org/
[4] http://www.eea.europa.eu/

divisions of USA The average intersection degrees of SA1, NSWMB, and CS are about seven, while that of SA1 is about 24. (CS; $36{,}702$ regions), and School Catchment Areas of USA (SC; $65{,}192$ regions). Real-4 consists of four datasets about species distribution and habitat from EEA: HEU($5{,}322$ regions; average intersection degree about 254), HMS($6{,}258$ regions; average intersection degree about 247), SEU ($10{,}061$ regions; average intersection degree about 467) and SMS ($11{,}613$ regions; average intersection degree about 474).

Real-1 and Real-2 are mainly used to illustrate the two factors that affect the qualified size of MA, while Real-3 and Real-4 show the scalability of this approach.

Result. Figure 5.6 shows the results of our experiments for Real-1 and Real-2. It is immediately clear from Figure 5.6 that the other algorithms in the figure dominate the complete QCN representation ('Complete' in the figures) in all cases. Figures 5.6(a) and (b) show for Real-1 the qualified size of MA, which is actually $n\bar{d}$ for CDC or $n\bar{d}/2$ for RCC8, and thus grows linearly in the number of regions n for both RCC and CDC relations. The qualified size of MA is consistently smaller than the grid or R-tree clustering indexes. Indeed, the qualified sizes of the grid and the R-tree clustering indexes quickly become prohibitively high as the number of regions increases for CDC (note that the y-axis is in $\log_{10}$ scale), although these two algorithms perform better for RCC8 than for CDC. This is probably because the clustering index mainly helps to distribute disjoint objects into different clusters, which can distinguish the clustering relation **DC** clearly from other relations for RCC8 but cannot distinguish well the clustering relations and non-clustering relations for CDC.

The results on Real-2 (Figures 5.6(c) and (d)) further show that MA is outperforming the other algorithms, and that it linearly depends on the average intersection degree. Taking a closer look, we can see that there are indeed sev-



(a) Real-1, RCC8 (in logarithmic scale)

(b) Real-1, CDC (in logarithmic scale)

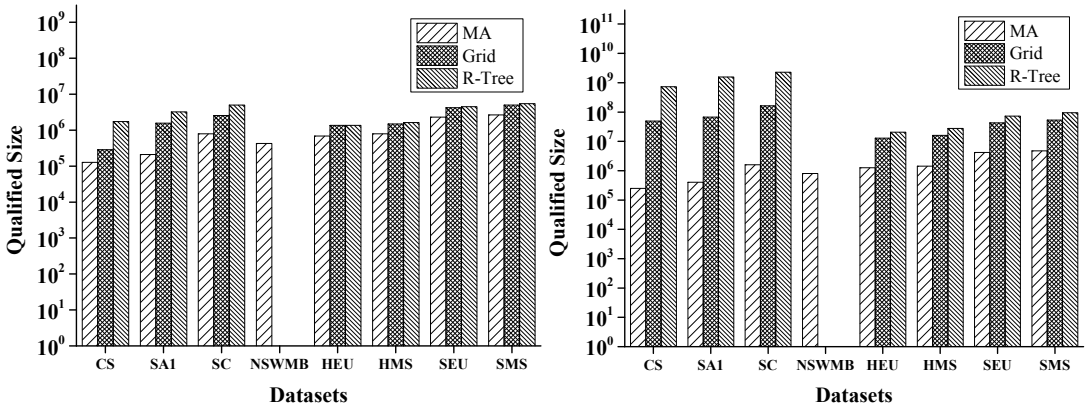
(c) Real-2, RCC8 (in linear scale)

(d) Real-2, CDC (in linear scale)

Figure 5.6: Qualified size variation with respect to the number of regions and the average intersection degree, on real-world dataset.

eral differences from the results for Real-1. First, the difference between the grid clustering index and the R-tree clustering index becomes much smaller for both RCC8 and CDC (noting that the y-axis uses a linear scale in Figures 5.6(c) and (d)). Second, the growth rates of the qualified size for the grid and the R-tree clustering indexes are not as high as the growth rates for Real-1. The first difference is due to the large number of regions intersecting with each other in Real-2. This makes it especially hard for both grid and R-tree clustering indexes to find good clustering structures. Consequently there is very little difference between the clustering powers of these two approaches. The second difference is due to poor performance of both algorithms. To be specific, as the average intersection degree grows while the number of regions is fixed, the qualified sizes for both grid and R-tree clustering indexes remain large.

For the larger datasets in Real-3 and Real-4, Figure 5.7 shows the results. In the results we omitted the complete QCN representation for all these large



(a) RCC8 (in logarithmic scale) (b) CDC (in logarithmic scale)

Figure 5.7: Qualified size comparison on large real-world datasets.

datasets, as it will have too large numbers that makes the comparison of the other methods unclear. From these results on large datasets, we can still observe that MA has the best results on all the datasets and has a better scala-

bility (for CDC we can see differences in order of magnitude). Note that for the dataset NSWMB in Real-3, both the grid and R-tree clustering indexes have exceeded the 16GB memory limit for RCC8 and CDC. This is another advantage of MA noticed in the experiments, i.e. it requires a much smaller memory size than the grid and the R-tree clustering indexes to construct the representation. In particular, it requires less than 1GB for all of the datasets, while the other two approaches cost more than 10GB for the large datasets in Real-3 and Real-4. We think this is due to the clustering index structure of these two approaches and the auxiliary record that is used to remove repeatedly calculated pairs of regions.

In summary, MA outperforms both of the grid and R-tree clustering indexes for CDC and RCC8, as well as the complete QCN representation for RCC8 or CDC. The advantage of MA is especially noticeable for CDC, because the clustering structures cannot distinguish well the clustering relations for CDC.

5.4 Query Support

We focus on the type of queries about checking the relation between two given regions. This type of queries can be regarded as the most fundamental one, as we have discussed in the introduction of the compact representation problem in Section 2.5.5. The essence of other types of queries, such as the one about finding all regions that satisfy a relation with a given region, involves inferring or obtaining the actual relation between any two regions, i.e. the query type that we focus on.

One factor that may affect the efficiency of query answering is the choice of data structures where the relations are stored. One should note that this choice is generally task dependent. For example, we could store the relations in a

relational database, in which case we can use both the variables and relations as identifiers. It has the advantage of being more flexible in the types of queries that are supported. One could also store the relations in a hash table for more efficient retrieval, which however might have scalability problem.

Note that it is possible to index the objects by using R-trees and other related data structures to improve retrieval efficiency (see e.g. [9, 61, 95]). Such techniques mainly focus on answering the type of queries about all pairs of regions that satisfy a given relation and assume that the relations between regions are explicitly available. We should note that such techniques of using R-trees are different from the R-tree clustering index, where they assume that the qualitative relations are already known and use data structures like R-trees for regions to filter out pairs of regions that could not satisfy the given relation. These techniques might be useful to further improve the efficiency of retrieving relations, e.g., by building spatial index on the MBRs. Currently we do not consider such optimisations, in order to focus on discussing how the representation itself could be helpful.

To answer the queries, we use both the stored RCC8/CDC relations and the MBRs of regions, and sometimes the MBRs of the connected components of the regions if the calculus is CDC. In general, to identify the relation between two regions, it runs as follows. First, we check if the relation is stored explicitly (e.g., for CDC we check if two MBRs intersect interiorly), if so then return it; otherwise use the MBRs to calculate the relation according to Propositions 5.1 and 5.4. In the analysis that follows, we focus solely on the more challenging CDC relation. For RCC8, both MA and the grid/R-tree clustering indexes only omit **DC** relations, and so the query performance for RCC8 is highly predicable and similar across all these methods.

For CDC, in practice, we do not need to strictly follow the procedure in-

duced by Proposition 5.4. Consider the example shown in Figure 5.4 again. To calculate the CDC relation from a to b, by $\delta(\mathsf{mbr}(a), \mathsf{mbr}(b)) = \{\mathsf{NW}, \mathsf{N}, \mathsf{NE}\}$, it is not difficult to see that a must have connected components say a_1 and a_2 such that $\mathsf{NW} \in \delta(a_1, b)$ and $\mathsf{NE} \in \delta(a_2, b)$. Hence we only need to check if there exists a connected component a_i of a such that $\mathsf{N} \in \delta(a_i, b)$. Even better, in this case, we only need to check if the x-projection of a

$$I_x(a) = \{x : (\exists y)(x, y) \in a\} \tag{5.1}$$

has non-empty intersection with the open interval (x_b^-, x_b^+), where

$$x_b^- = \inf\{x : (\exists y)(x, y) \in b\} \quad \text{and} \quad x_b^+ = \sup\{x : (\exists y)(x, y) \in b\} \ . \tag{5.2}$$

The following observation can be used to further simplify the procedure.

Proposition 5.9. *Given two possibly disconnected regions a, b such that $(\mathsf{mbr}(a))^\circ \cap (\mathsf{mbr}(b))^\circ = \varnothing$, if $\delta(\mathsf{mbr}(a), \mathsf{mbr}(b))$ contains at most two single tile relations, then $\delta(a, b) = \delta(\mathsf{mbr}(a), \mathsf{mbr}(b))$.*

This is the case when for example $\delta(\mathsf{mbr}(a), \mathsf{mbr}(b))$ is a single tile relation other than O, which includes not only the CDC clustering relations but also some others like N and W.

Based on the above observations, now we present the query answering method for CDC based on the representation obtained by MA, denoted by MAQ. Write R_{12} for the CDC relation from o_1 to o_2 (i.e. o_2 as the reference object) and S_{12} for the CDC relation from $\mathsf{mbr}(o_1)$ to $\mathsf{mbr}(o_2)$, where $\mathsf{mbr}(o_1)$ and $\mathsf{mbr}(o_2)$ are the MBRs of o_1 and o_2, respectively. If $(\mathsf{mbr}(o_1))^\circ \cap (\mathsf{mbr}(o_2))^\circ \neq \varnothing$, then we know the CDC relation is explicitly stored. Otherwise, we have the following conclusions.

- If o_1 is a connected region, then R_{12} is the same as S_{12} by Proposition 5.2.
- If o_1 is a disconnected region, then R_{12} and S_{12} can only be 1-tile, 2-tile, or 3-tile relation. We only need to consider the two cases below.
  - Case 1: if S_{12} is a 1-tile or 2-tile relation, then R_{12} is the same as S_{12} by Proposition 5.9.
  - Case 2: if S_{12} is a 3-tile relation, then it can only be one of these four cases: $\{\mathsf{NW}, \mathsf{N}, \mathsf{NE}\}$, $\{\mathsf{NW}, \mathsf{W}, \mathsf{SW}\}$, $\{\mathsf{NE}, \mathsf{E}, \mathsf{SE}\}$, and $\{\mathsf{SW}, \mathsf{S}, \mathsf{SE}\}$. The difference between R_{12} and S_{12} lies only in the presence of the 'middle' tile name (e.g. N is the middle tile name of $\{\mathsf{NW}, \mathsf{N}, \mathsf{NE}\}$, and E for $\{\mathsf{NE}, \mathsf{E}, \mathsf{SE}\}$) in them. Therefore, we only need to check if there is a connected component o_1^i of o_1 that intersects with the middle tile interiorly, and if so then $R_{12} = S_{12}$, otherwise R_{12} is S_{12} excluding the middle tile.

For Case 2 above, it can be accomplished by simply checking the intersection of $I_x(o_1)$ (or $I_y(o_1)$) and $(x_{o_2}^-, x_{o_2}^+)$ (or $(y_{o_2}^-, y_{o_2}^+)$) as specified in (5.1) and (5.2). In particular, for each connected component o_1^i of o_1, we check the projection of the rectangle $\mathsf{mbr}(o_1^i)$ on the x/y-axis and $(x_{o_2}^-, x_{o_2}^+)$ (or $(y_{o_2}^-, y_{o_2}^+)$). For example, for $S_{12} = \{\mathsf{NW}, \mathsf{N}, \mathsf{NE}\}$, we check the projection of $\mathsf{mbr}(o_1^i)$ on the x-axis and $(x_{o_2}^-, x_{o_2}^+)$. If they intersect, then $\mathsf{mbr}(o_1^i)$ must have an interior intersection with the middle tile N, because $\mathsf{mbr}(o_1^i)$ can only lie in the area covered by tiles NW, N, and NE.

For the grid and R-tree clustering indexes, it should be noted that in [44] and [43] Fogliaroni et al. did not consider the mechanism to answer the queries discussed here (i.e. the ones about finding the CDC relation between two given objects). Nevertheless, there are many possible ways to answer this type of queries for the grid and R-tree clustering indexes. Specifically, when

the MBRs are available, we can exploit the following strategy for obtaining the CDC relation $\delta(o_1, o_2)$.

1. First compute the CDC relation $\delta(\mathsf{mbr}(o_1), \mathsf{mbr}(o_2))$ from $\mathsf{mbr}(o_1)$ to $\mathsf{mbr}(o_2)$;
2. If the relation is one of the CDC clustering relations (NW, NE, SW, SE), then $\delta(o_1, o_2) = \delta(\mathsf{mbr}(o_1), \mathsf{mbr}(o_2))$;
3. Otherwise, the relation between the regions must have been previously computed and stored in the resulting representation.

The difference between this strategy and MAQ mainly lies in the case where the MBR relation is not a clustering relation and at the same time it does not contain the tile name O (that is, the two MBRs do not intersect interiorly). In such case, this strategy will retrieve the stored relation while MAQ will use the MBRs to compute the relation. We denote this query answering strategy by GQ for the grid clustering index and by RTQ for the R-tree clustering index.

5.4.1 Empirical Evaluation

In Experiment 2, the performance of answering a basic spatial query was tested using the different representations from MA, the grid clustering index, the R-tree clustering index, the complete QCN (denoted by the word "*Complete*"), and using direct geometric computation of relations between given objects (denoted by "Direct"). The specific spatial query used here is to find the CDC/RCC8 relation between two given objects. In the analysis that follows, we focus solely on the more challenging CDC relation. For RCC8, the MBR-based algorithm, and the grid and R-tree clustering indexes only omit the **DC** relation, and so the RCC8 query performance is highly predicable and similar across all these methods.

To answer the query for the representation obtained by MA, we use the aforementioned query answering strategy MAQ. Based on MAQ, direct computation can be optimised in the case where the MBRs do not intersect interiorly, by exploiting exactly the same steps of MAQ. In the following experiments, we will always apply this optimisation to direct computation.

For the representation obtained by the grid and the R-tree clustering indexes, we will use the strategies GQ and RTQ to answer the queries respectively. Moreover, in the experiment, to make MAQ and GQ/RTQ comparable, MAQ will first check if two MBRs are in a clustering relation. This will only slightly increase the query time. For direct computation, when it comes to check if two geometries intersect, we will first use the MBRs of the two geometries to pre-test the possibility of intersection.

We will assume that the calculated relations, MBRs and geometric information are all available in memory. For each method and dataset, the calculated relations are stored in a database that is hash indexed using the identifiers of geometries as keys, e.g. the key for relation R_{ij} is $i \times N + j$, where N is a sufficiently large integer. Here the MBRs serve as an index for the relations stored in the database. The experiments have been done on a computer running Ubuntu 14.04, with an Intel® Core™-i3 1.6GHz CPU and 16GB memory.

We tested answering queries on $10{,}000$ random pairs in each of the largest dataset (USA-adm2) in Real-1 and the one with highest average intersection degree in Real-2 (called Real-2.5), as illustrations of the performance of MAQ. The $10{,}000$ pairs are chosen by randomly sampling in the set of all pairs of regions. For the larger datasets in Real-3 and Real-4, the representations from the grid/R-tree clustering indexes and the complete QCN representation can hardly be fitted into the memory, and hence we did not include them here. Later in the next section, we will show the query answering performance for

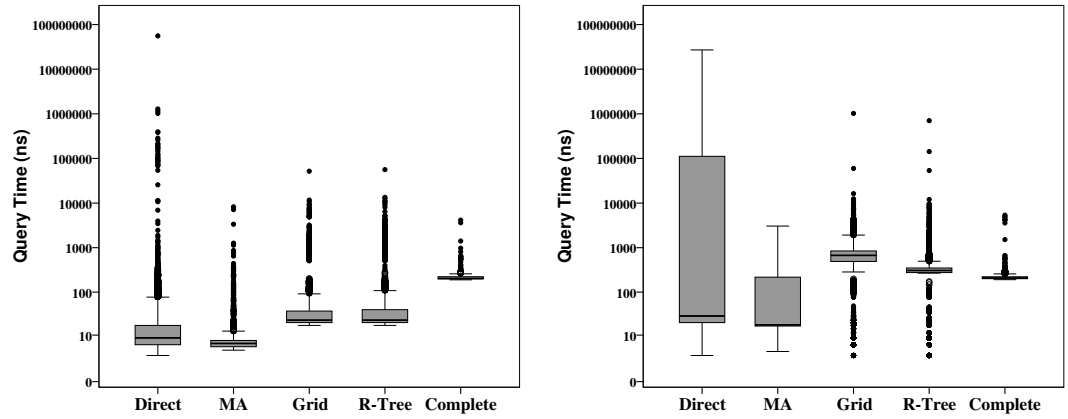


Figure 5.8: (a) Query times on the dataset USA-adm2 in Real-1. (b) Query times on the dataset with highest average intersection degree in Real-2 (i.e. Real-2.5). Note that '$\circ$' represents values that lie more than 1.5 box lengths from the hinge of the box.

MAQ on datasets in Real-3 and Real-4.

Result. As we can see from Figure 5.8, answering queries by MAQ exhibits promising performance. Compared with the two extremes, it is at least as efficient as direct computation from the geometry and is substantially faster than retrieval from the complete QCN (note that all the relations are stored in an in-memory database). Like the query answering strategies by the grid and R-tree clustering indexes, MAQ represents a compromise that can support more efficient queries than either of these two extremes. In particular, for USA-adm2, the median query time of MAQ is respectively about $28\%, 73\%, 73\%$, and 97% lower than that of direct computation, GQ, RTQ and the complete QCN representation; for Real-2.5, the median query time of MAQ is respectively about $38\%, 97\%, 94\%$, and 91% lower than that of the other methods. Hypothesis tests also confirmed the visual impression from Figure 5.8, that on average MAQ can be more efficient than all the other approaches (Mann-Whitney U test, significant at the 5% level). However, it should be noted that such statis-

tical significance in this case is only a guide, as a large number of samples and different distributions can be misleading towards comparisons of differences. The difference between USA-adm2 and Real-2.5 is that the former contains a lot of regions that are disjoint. Therefore, more of the CDC relations can be calculated from the MBRs of regions in USA-adm2, rather than to retrieve from the database or calculate from geometric representation. This explains the difference between the performance on these two datasets of all methods except the complete QCN representation. In summary, the result indicates that the representation obtained by MA is a useful alternative in practice to support efficient answering of queries on the qualitative spatial representation, in addition to reducing construction time and size of the representation.

Discussion. Even though the direct computation shows a performance that is good to some extent, it is worth noting that this method is not always efficient for answering queries. The reason is as follows. There are many "degenerate" cases for the direct computation. In fact, in the experiment on Real-2.5, for the direct computation, about $3{,}000$ instances (30% of all the tested ones) have query answering time of more than $10{,}000$ns, compared with the case of MAQ, which has no instances beyond time $10{,}000$ns.

Sometimes the calculated relations would be stored in hash tables in memory, the operation to access a relation will be much faster than the case in the experiment. When the hash table is not very large, for these methods, the more relations that are explicitly stored, the faster the query answering will be. In this case, the performance of the complete QCN will be the best, followed by GQ and RTQ, and then MAQ. However, this storage mechanism will not be scalable for the complete QCN, whose size is quadratic to the number of objects. This makes us choose the database storage instead for the experiment.

In some other cases, the calculated relations would be stored in a database

on hard disk rather than in memory. In this case, all the methods except for the direct computation will take more time to answer the query, as reading data from hard disk is slower than from memory. The MBR query method will require fewer reading operations than the others because it stores fewer relations. Another possibility is that the MBRs and geographic information are also stored on hard disk. In this case, all the methods except retrieving from the complete representation will require more reading operations, as they would probably use the MBRs and geometric information to calculate the relations. The exact impact of storing relations on disk is difficult to measure, as performance will crucially depend on optimisations by the database and operating system, such as bundling several queries to the database or by caching some frequent queried relations.

We should also note that although query answering by the MBR query method on average equals or outperforms all the other approaches, it is possible to find or construct degenerate cases, where the performance of this method could be worse than these alternatives. In particular, in the case where the relation of MBRs contain three tile names and these MBRs do not intersect in their interiors, the MBR query method might lead to inefficient queries. In such cases, answering queries by using the MBR query method needs to check if the 'middle' tile name is valid in the real CDC relation. This in turn involves checking intersection of the tile and the MBRs of the connected components in the primary object. For example, in Figure 5.4, the MBRs of region a and region b are in CDC relation with three tile names $\{\mathsf{NW}, \mathsf{N}, \mathsf{NE}\}$. Note that a contains many connected components and none of them intersect with the middle tile N. In this case, to compute the actual CDC relation between a and b using the MBR approach, the query will probably need to check the MBRs of *all* the connected components of a.

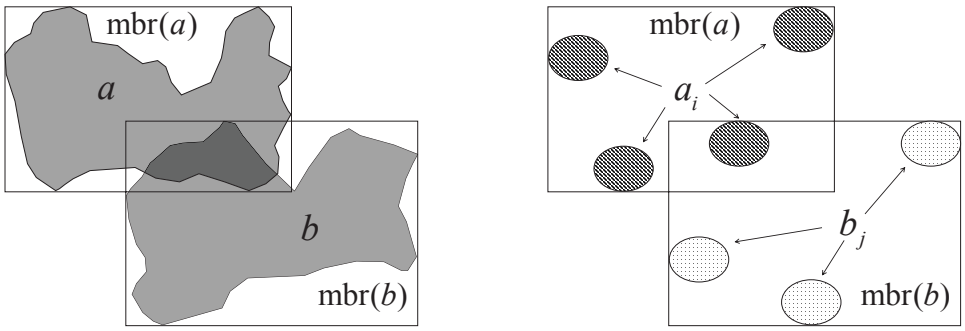
However, such degenerate cases seem to be rare in practice. As we have seen, on average answering queries based on the MBR approach is the most efficient one of all the alternatives tested. Note that on average, each region in USA-adm2 contains 3.5 connected components and in Real-2.5 the number is 34. This means in practice the validity of the "middle" tile name can usually be confirmed after checking only a few MBRs of connected components. Moreover, the performance on the degenerate cases can be optimised, such as by spatially indexing the MBRs of connected components to reduce the number of tests for intersection.

Finally, we note that all the methods might be further optimised in real-world applications. For example, the MBRs of connected components of the regions can be indexed using advanced spatial indexing techniques such as R-Tree [95] to reduce computation when checking CDC relations. Here we are more interested in analysing the performance of the MBR query method itself. Thus, to be clearer about how the MBR query method performs, we were not applying other optimisations, and the experiments here are just illustrations of the performance of the MBR query method, to show it is feasible in practice.

5.5 Derivatives of the MBR-Based Approach

The qualified size of MA can be further reduced. In this section, we discuss three derivatives of MA that further reduce the qualified size of MA.

For the first technique, denoted by MC, to decide which relations need to be explicitly stored, it first applies MA to $\mathcal{D}$, and then for the stored relations, it compares the CDC/RCC8 relation $R_{a,b}$ between two regions a and b with the CDC/RCC8 relation $R_{\mathsf{mbr}(a),\mathsf{mbr}(b)}$ between the MBRs of the two regions $\mathsf{mbr}(a), \mathsf{mbr}(b)$. If the relation $R_{a,b}$ is the same as $R_{\mathsf{mbr}(a),\mathsf{mbr}(b)}$, e.g. Fig-



(a) A case where MC works. (b) A case where MD works.

Figure 5.9: Illustration of cases where MC or MD will not store the relation while MA will.

ure 5.9(a), then we remove $R_{a,b}$ from the storage. This technique would work well if $R_{a,b} = R_{\mathsf{mbr}(a),\mathsf{mbr}(b)}$ holds for many pairs of regions. In the query stage, to find the correct relation between a and b, it then suffices to return the relation that is stored, if it is available, and to calculate $R_{\mathsf{mbr}(a),\mathsf{mbr}(b)}$ otherwise.

For the second technique, denoted by MD, it further reduces the qualified size of MA, by removing any **DC** relation $R(a, b)$ in the representation that satisfies the following condition: for every connected component a_i of a and for every connected component b_j of b, we have that $\mathsf{mbr}(a_i)$**DC**$\mathsf{mbr}(b_j)$ (e.g. see Figure 5.9(b)). To efficiently check this condition, we subsequently check the following conditions:

1. for every connected component a_i of a, $\mathsf{mbr}(a_i)$**DC**$\mathsf{mbr}(b)$;
2. for every connected component b_j of b, $\mathsf{mbr}(a)$**DC**$\mathsf{mbr}(b_j)$.
3. for every connected component a_i of a that does not satisfy the first condition and for every connected component b_j of b that does not satisfy the second condition, $\mathsf{mbr}(a_i)$**DC**$\mathsf{mbr}(b_j)$.

The removed **DC** relation $R_{a,b}$ can be retrieved by using the MBRs of the connected components of the regions:

1. if $\mathsf{mbr}(a)$**DC**$\mathsf{mbr}(b)$ then $R_{a,b} =$ **DC**;

2. otherwise, if one of the above three conditions are satisfied then $R_{a,b} =$ **DC**;

3. if none of the above holds, then the relation must have been stored.

This method will work well if the MBRs of the connected components approximate the disjointness of the regions well. Note that this method comes at the cost of efficient query answering than MA and MC, since a larger number of relations between the MBRs may need to be checked. Based on MD, there could be techniques dealing with relations other than **DC**. The idea would be similar, but involves many other details and we would not consider them here.

The third technique, denoted by MM, combines MC and MD. Based on the representation obtained by MC, it further removes the relations satisfying the conditions for MD. For MM, the removed **DC** relation $R_{a,b}$ can be retrieved as follows

1. if $\mathsf{mbr}(a)$**DC**$\mathsf{mbr}(b)$ then $R_{a,b} =$ **DC**;

2. otherwise if one of the three conditions for MD are satisfied then $R_{a,b} =$ **DC**;

3. otherwise if $R_{a,b}$ is stored, then return the stored relation;

4. if none of the above holds, then we know $R_{a,b} = R_{\mathsf{mbr}(a),\mathsf{mbr}(b)}$ by the specification of MC.

Table 5.2: Reduction rates of the qualified size for CDC relations of MC over MA.

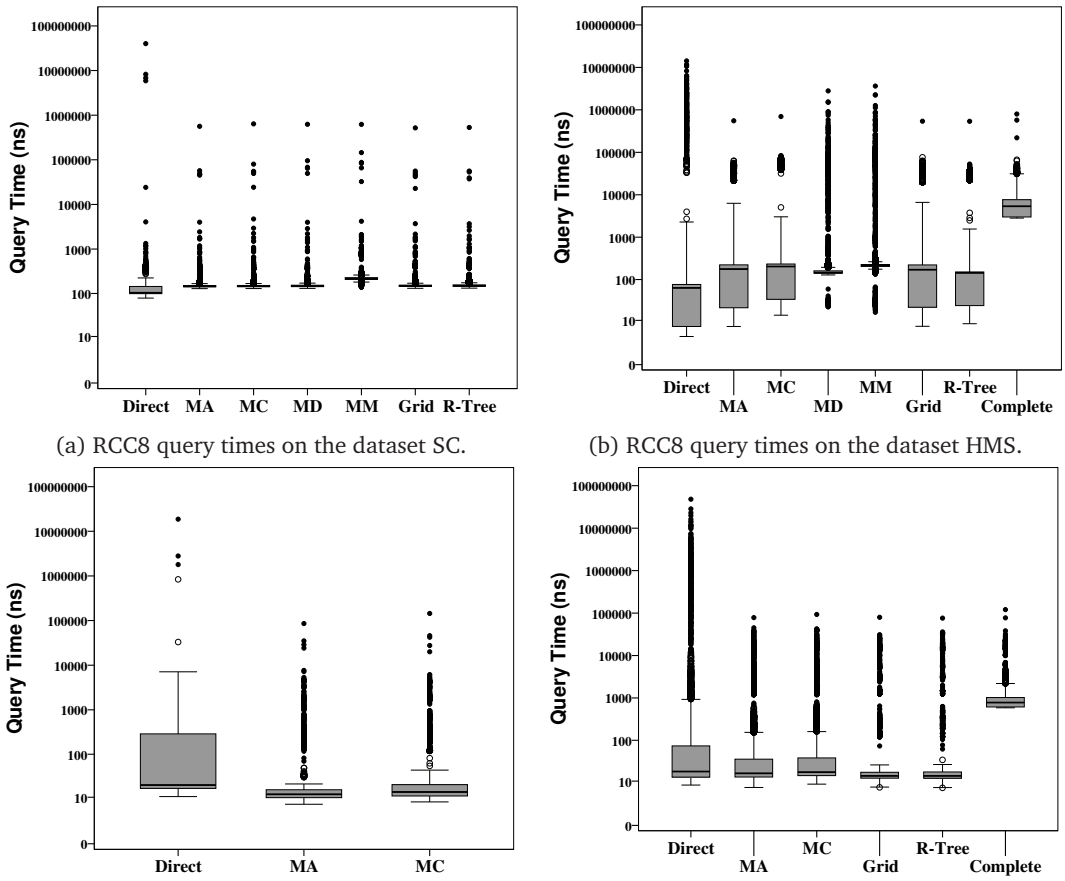
Dataset	SA1	NSWMB	CS	SC	HEU	HMS	SEU	SMS
MC	0.000%	0.000%	0.000%	2.596%	0.232%	0.189%	0.258%	0.231%

Table 5.3: Reduction rates of the qualified size for RCC8 relations of the derivatives of MA over MA.

Dataset	SA1	NSWMB	CS	SC	HEU	HMS	SEU	SMS
MC	3.969%	5.788%	3.000%	34.223%	9.752%	10.205%	10.371%	10.898%
MD	0.260%	0.000%	0.324%	3.390%	44.604%	44.444%	40.043%	40.883%
MM	4.229%	5.788%	3.324%	37.613%	54.357%	54.649%	50.415%	51.781%

Experiments (Table 5.2 and Table 5.3) on these approaches show that for the RCC8 relations on some datasets there could be a large reduction, while for CDC relations the reduction is relatively small. For CDC, this is probably due to the good performance of MA, where a large portion of the relations that coincide with the relations of MBRs have been identified. For RCC8 in Table 5.3, we observe that higher reduction rates of MC mostly occur for method MD and MM in the datasets that have a relatively higher average intersection degree and a larger number of connected components for each region (i.e. HEU, HMS, SEU, SMS). This is probably because in such case a lot of regions are in **DC** relations but their MBRs are not. Moreover, the dataset SC has much higher reduction rates than the others in Real-3, especially for the methods MC and MM. Note that the average intersection degree of this dataset is relatively large and in the meantime the average number of connected components is relatively small. These might lead to that the relations of the regions are likely to be the same as the relations of the MBRs of the regions or the MBRs of the connected components. Finally, note that for the datasets in Real-4, the high reduction rates are mainly due to MD, which, compared to MA, only further removes some **DC** constraints. Therefore, although the reduction rates on these datasets are high, there are still a large number of non-**DC** constraints being stored.

Next, let us have a look at the query answering times. Figure 5.10 compares the time of answering queries on the datasets SC and HMS (which have high



(a) RCC8 query times on the dataset SC.

(b) RCC8 query times on the dataset HMS.

(c) CDC query times on the dataset SC.

(d) CDC query times on the dataset HMS.

Figure 5.10: Comparison of the performance of query answering with MC, MD, MM and other methods on the datasets SC and HMS.

reduction rates) by these approaches with previously discussed approaches. It should be noted that the representations of some approaches exceeded our memory limit (16GB) and thus we did not show their query times in the figure. These include the complete QCN representation ("Complete") on the dataset SC for both RCC8 and CDC, and the grid/R-tree clustering indexes ("Grid" and "R-Tree") on the dataset SC for CDC.

From the results, we can see that in all the cases, MC has a similar performance as MA and the grid/R-tree clustering indexes, which is probably due to their similar query strategies, i.e. first checking if MBRs disjoint and then retrieve relation from database or calculate the relation from MBRs. Also, note that MD and MM have a worse performance on dataset HMS than on SC. This indicates that the reduction in storage space results in longer query times for MD and MM. Finally, the direct computation ("Direct") has a smaller average query time than the other approaches on SC and HMS for RCC8. This is not surprising, because, especially for RCC8, there are a large proportion of cases where the optimisation of MA can be applied to Direct (e.g., for RCC8, Direct can pre-test MBR intersection), and on these datasets direct computation sometimes can be more efficient than I/Os of databases applied in all the other methods. On the other hand, by comparing the results of Direct on HMS and SC, we see that the number of long query times of Direct increases when there is an increase of the number of cases where the optimisation of MA can not be applied (i.e. when MBRs have intersections).

5.6 Summary and Discussion

We have seen that MA and its derivatives have promising performance for encoding topological and directional information compactly, while retrieving the

relation of any given pair of variables is still efficient. Moreover, they have provided an integrated representation for both the RCC8 and CDC information, by using MBRs. This is desirable when we want to build a qualitative database of different aspects of spatial and temporal information, as it saves storage space simultaneously for different information. Finally, the MBR information is usually available in a spatial database, which would make it easy to incorporate the compact representation with the current database.

In the previous chapter, we have discussed the prime subnetwork technique for simplifying the complete QCN representation. By removing redundant constraints, this technique will reduce the number of constraints in storage. However, as have been observed before, this technique has several problems to be applicable for the task here. First and foremost, it currently does not provide efficient enough strategy to answer the query about the relation between two given regions/variables. The only way to answer such query by using the prime subnetwork representation would be using the qualitative reasoning techniques such as PC, PPC, and DPC discussed in Chapter 3. Although these techniques are efficient in terms of qualitative reasoning, they cost too much time to answer the queries, especially when the number of regions/variables becomes large (note that the worst case time complexity for these techniques is $O(n^3)$). For example, even for datasets in Real-2 with only about 600 variables, it takes 10^7ns to retrieve the relation that removed in the prime subnetwork representation. Also, it is not applicable to the CDC information, which is the main focus of MA. Therefore, we do not consider it as a competitive alternative for the compact representation problem.

Another technique to simplify the complete QCN representation is specifically for the RCC8 information. This technique, called *Non-**DC***, removes all the **DC** relations in the complete QCN, and assume that if a relation for two regions

is not stored then it is **DC**. The resulting representation will have less stored relations than the one obtained by MA and MD, as well as the one by the grid or the R-tree clustering indexes. This is because the only type of relations that the latter techniques do not store is **DC** relations. Nevertheless, Non-**DC** techniques have several shortcomings compared to MA. Obviously, it does not work for CDC information. Also, because Non-**DC** assumes that a relation not stored is always **DC** (similar to the closed world assumption), sometimes it might return the incorrect relation. For example, when some stored relations were lost due to some unexpected error without being noticed, the Non-**DC** method will still assume these records are **DC** relations. On the other hand, MA (and MD) will report unknown or error in this case, because they first check if the MBRs have intersection to see whether the relation should have been stored. Note that however MC and MM have the same problem of returning the wrong relation, as they assume the relation that is not found in the records can always be calculated from the MBRs.

We should note that MA and its derivatives can only deal with the case where the input is a set of geometric shapes. In real-world applications, there are also many cases where the input is in the form of qualitative relations, such as text descriptions about places in Wikipedia (e.g. the description of Darling Harbour in Chapter 1). Also, as mentioned above, for RCC8 relations Non-**DC** will dominate the current techniques, including MA and MD. On the other hand, Non-**DC** has the problem of reporting wrong relations in certain cases, as discussed above, and there are datasets with a large number of non-**DC** relations (e.g. the datasets in Real-4) where Non-**DC**, as well as MA and its derivatives, cannot perform very well. Therefore, in the next chapter, we propose another technique to compactly encode the RCC8 relations while supporting efficient relation retrieval for any two regions or variables.

Chapter 6

Compact Representation: Encoding with Rectangles

6.1 Introduction

In the previous chapter, the main idea is to use MBRs to approximate the regions, such that some of the relations can be unambiguously inferred from the MBRs. It is based on the observation that many real-world datasets have relatively small "average intersection degree", which counts the number of MBRs intersecting a given one in the dataset on average. However, in applications, there are also cases where (i) the average intersection degree is quite large (ii) for some regions we may only know how they qualitatively relate to other regions, without access to precise boundaries (see, e.g., [107, 124, 126]). The MBR-based approach MA (and its derivatives) will be less useful in these cases, since its performance will be worsened when the average intersection degree becomes larger and it relies on the polygon representation of the regions.

Therefore, in this chapter, we will present another approach that uses axis-aligned rectangles to compactly encode the RCC8 relations between a set of

regions, such that the relation between any two regions can be retrieved efficiently. These rectangles usually are not the MBRs of the regions. The scope of this chapter is restricted to basic RCC8 relations.

In Section 6.2 we give an algorithm to construct a rectangular pseudo solution for an RCC8 scenario and show how to efficiently retrieve the relation between two entities with the pseudo-solution representation. Section 6.3 discusses different implementation details of the algorithm to construct pseudo-solutions. We empirically evaluate the effectiveness of our method on large real-world datasets and analyse its performance in Section 6.4.

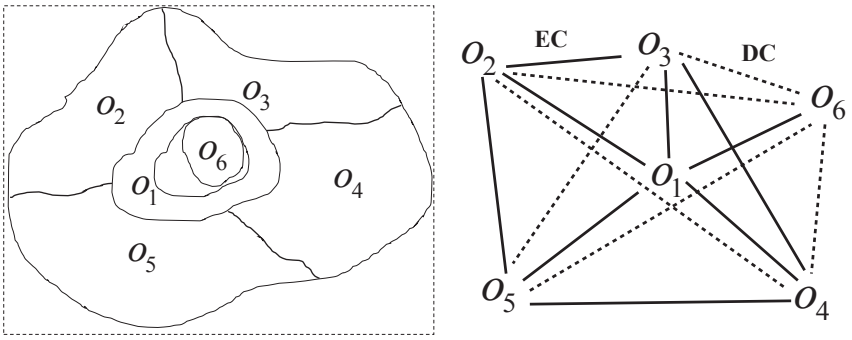
Bibliographic Note. The work reported in this chapter is based on the joint work with Steven Schockaert and Sanjiang Li, following the research on compact representation in the previous chapter. In the beginning of this research, Sanjiang Li, Steven Schockaert, and I discussed about the possibility to generate simple shapes for compact representation. Later, Steven Schockaert and I developed the idea of pseudo-solution, with intriguing proposals from Sanjiang Li. With the help of Steven Schockaert and Sanjiang Li, I devised the algorithm, implemented it, and conducted the empirical evaluations. Most of the contents in this chapter were first presented in the following publication. The algorithm used here has been adjusted a little, by changing the way to construct rectangles, as discussed in Section 6.2.1.

- Zhiguo Long, Steven Schockaert, and Sanjiang Li: Encoding Large RCC8 Scenarios Using Rectangular Pseudo-Solutions, KR, 2016, pp. 463–472.

6.2 Pseudo-Solutions

Recall that a QCN $\mathcal{N}$ can be seen as a set of constraints between several variables. A solution of $\mathcal{N}$ is an assignment of entities in a domain to the variables

such that the constraints between all the variables are satisfied by the entities. For example, Figure 6.1(a) gives a set of regions and there is a corresponding QCN about the RCC8 relations between these regions in Figure 6.1(b). Then the regions naturally give a solution of the QCN as every constraint in the QCN is satisfied by these regions.

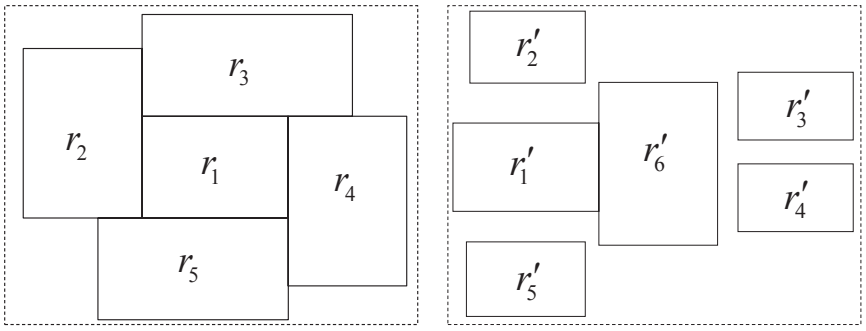


(a) V (b) $\mathcal{N}$

Figure 6.1: A set of regions $V = \{o_1, \ldots, o_6\}$ and the corresponding RCC8 QCN $\mathcal{N}$.

For the QCN in the above example, when we restrict the domain to rectangles, there will be no solution of the corresponding $\mathcal{N}$, although the RCC8 QCN $\mathcal{N}$ itself is consistent. This is because, when the rectangles $r_1, \ldots, r_5$ satisfy the relations between $o_1, \ldots, o_5$, we cannot find a rectangle r_6 for o_6 that satisfies all the relations between o_6 and $o_1, \ldots, o_5$. Figure 6.2(a) illustrates this circumstance, where there is no rectangle that can be in **EC** relation with r_1 while in **DC** relation with the other rectangles.

Nevertheless, by removing enough number of constraints from $\mathcal{N}$, we can always find a rectangular solution of the resulting QCN. For example, by removing the constraints between $o_1, \ldots, o_5$, we can find another set of rectangles that satisfy the other constraints, as shown in Figure 6.2(b). Here, we say a QCN $\mathcal{N}$ can be *weakened* by removing one or several constraints from $\mathcal{N}$. Based on weakened QCNs, we can then define a *pseudo-solution* of $\mathcal{N}$.



(a) S_0 (b) S_1

Figure 6.2: An illustration of a pseudo-solution of the spatial scenario in Figure 6.1.

Definition 6.1. Given a QCN $\mathcal{N}$ and the set of variables is V, a *pseudo-solution* of $\mathcal{N}$ is a sequence $\mathcal{L} = \langle S_0, \ldots, S_k \rangle$ of assignments to $V_0, \ldots, V_k \subseteq V$ such that there exists a sequence of progressively weakened networks $\langle \mathcal{N}_0, \ldots, \mathcal{N}_{k+1} \rangle$ that satisfy:

- $\mathcal{N}_0 = \mathcal{N}$, $\mathcal{N}_{k+1} = \varnothing$, and $\mathcal{N}_i = \mathcal{N} \setminus (\bigcup_{j=0}^{i-1} \mathcal{N}|_{V_j})$ $(1 \leq i \leq k+1)$;
- S_i is a partial solution of $\mathcal{N}_i$ satisfying the constraints in $\mathcal{N}_i|_{V_i}$ for every $0 \leq i \leq k$.

Note that if $\mathcal{L}$ is a pseudo-solution, then every constraint in $\mathcal{N}$ is satisfied by at least one of the partial solutions in $\mathcal{L}$. We can easily verify this for the pseudo solution given by the rectangles in Figure 6.2. A pseudo-solution is not necessarily a solution of $\mathcal{N}$, and the QCN $\mathcal{N}$ might not even be consistent. However, as we will see later, a pseudo-solution $\mathcal{L}$ allows us to retrieve the basic RCC8 relation between any two variables from $\mathcal{N}$. Here, as in the example, we are particularly interested in complete basic QCNs over RCC8 (i.e. RCC8 scenarios) and *rectangular* pseudo-solutions, which consist of partial solutions that assign axis-aligned rectangles to variables. The reason we consider axis-aligned rectangles is that we can easily calculate the RCC8 relations be-

Table 6.1: Correspondence of basic RCC8 relations and RA relations.

DC	$\{b, bi\} \otimes \star \cup \star \otimes \{b, bi\}$
EC	$\{m, mi\} \otimes (\star \setminus \{b, bi\}) \cup (\star \setminus \{b, bi\}) \otimes \{m, mi\}$
PO	$\{o, oi\} \otimes (\star \setminus \{b, bi, m, mi\})$ $\cup(\star \setminus \{b, bi, m, mi\}) \otimes \{o, oi\}$ $\cup\{d, s, f\} \otimes \{di, si, fi\} \cup \{di, si, fi\} \otimes \{d, s, f\}$
TPP	$\{s, f\} \otimes \{d, s, f, eq\} \cup \{d, s, f, eq\} \otimes \{s, f\}$ $\cup d \otimes eq \cup eq \otimes d$
NTPP	$d \otimes d$
EQ	$eq \otimes eq$

tween them, which will be helpful for constructing a compact representation that supports efficient retrieval of RCC8 relations.

6.2.1 Constructing Pseudo-Solutions

To construct a rectangular pseudo-solution, we need to somehow transform the RCC8 scenario into QCNs where we can easily construct rectangular partial solutions. There are three ways for that. The first one is to build a correspondence between the RA relations and the basic RCC8 relations. The second one is to transform the basic RCC8 relations into IA relations. The third one is to seek help from the PA relations.

Correspondence between RA and RCC8

As we are constructing *rectangular* pseudo-solutions, the calculus RA, which can model the relations between rectangles, is the first consideration for transforming the basic RCC8 relations. However, there are several difficulties. First, we have seen in the example that a consistent RCC8 scenario might not have a rectangular solution, so the resulting RA QCN might not have a solution. This is not the hardest problem, and we can use the idea of pseudo-solution

to tackle it. The second difficulty is about the efficiency. When transforming an RCC8 basic relation into an RA relation, the resulting RA relation might be outside the largest known tractable subclass of RA (i.e. the strongly-preconvex subclass) identified by Balbiani et al. [6]. See Table 6.1 and also [95, Figure 4] and [76, Figure 7] for the correspondence of RA relations and basic RCC8 relations. This suggests that it might be NP-hard to determine if a basic RCC8 QCN has a rectangular solution and it would be inefficient to construct rectangular pseudo-solutions from RA QCNs.

Correspondence between PA and RCC8

We can decompose the corresponding RA relation of a basic RCC8 relation into several RA relations that are *PA representable*. An IA relation ρ is *PA representable* or *pointisable* [72, 122] if there exists a PA network $\mathcal{N}$ over variables $\{x_1^-, x_1^+, x_2^-, x_2^+\}$ such that $(x_1^- < x_1^+)$ and $(x_2^- < x_2^+)$ are in $\mathcal{N}$ and ρ is identical to the solution set $\mathsf{sol}(\mathcal{N})$ of $\mathcal{N}$, i.e.

$$\rho = \{([a^-, a^+], [b^-, b^+]) \mid \langle a^-, a^+, b^-, b^+ \rangle \in \mathsf{sol}(\mathcal{N})\}.$$

We say that an RA relation is PA representable if it is the Cartesian product of two IA relations that are PA representable. A *PA representation* of a basic RCC8 relation then can be defined accordingly.

Definition 6.2. Let R be a basic RCC8 relation. Suppose $\mathcal{N}$ is a PA QCN for variables $V = \{x_1^-, x_1^+, x_2^-, x_2^+, y_1^-, y_1^+, y_2^-, y_2^+\}$ such that $(x_i^- < x_i^+)$ and $(y_i^- < y_i^+)$ are in $\mathcal{N}$ for $i = 1, 2$. We say that $\mathcal{N}$ is a PA representation of R if there is an RA relation contained in R that is PA representable by $\mathcal{N}$, i.e. identical to the solution set of $\mathcal{N}$.

Since a PA representation is actually a PA QCN, a solution of it can be

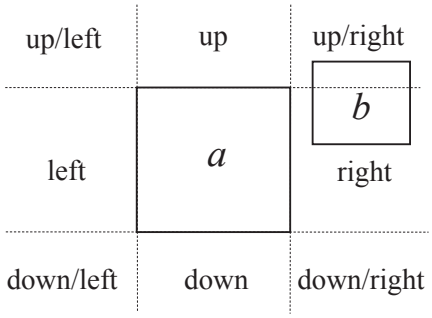


Figure 6.3: Two rectangles a and b in **DC** relation, where the corresponding PA representation of b is to the "right" of that of a.

considered as the endpoints of two rectangles. For example, Figure 6.3 shows two rectangles a and b whose endpoints form a solution of a PA representation of **DC** $\{(x_1^- < x_2^-), (x_1^- < x_2^-), (x_1^- < x_2^-), (x_1^- < x_2^-)\}$ (the relations for y-coordinates are omitted since they are universal relations). Generally for a basic RCC8 relation R, we say that a set of its PA representations *cover* R, if, for any two rectangles satisfying the RCC8 relation, the endpoints of them also satisfy one of the PA representations.

In addition, if we consider a as the reference object, then b is on the "right" of a. In fact, all pairs of rectangles (a, b) whose endpoints form a solution of this PA representation will satisfy that b is on the "right" of a. Then we say that the PA representation corresponds to the *relative position* "right". It may have been noticed that for the "right" PA representation of **DC**, some rectangles might also be in relative position "up" or "down", as shown in Figure 6.3. However, here we will associate this PA representation with "right", as there is another PA representation corresponding to "up" that covers this case. Similarly, we can characterise other PA representations of **DC** with relative positions, as well as those of **EC**, **TPP**/**TPP**$^{-1}$, and **PO**.

Table 6.2 shows a choice of PA representations that cover the basic RCC8 relations. Note that these PA representations are not the only choices that

cover the basic RCC8 relations. In Table 6.2, we use "r", "l", "u", and "d" to specify the relative positions of each PA representation. Note that sometimes a PA representation can corresponds to several relative positions.

Table 6.2: Correspondence of basic RCC8 relations and PA representations.

	(x_1^-, x_2^-)	(x_1^-, x_2^+)	(x_1^+, x_2^-)	(x_1^+, x_2^+)	(y_1^-, y_2^-)	(y_1^-, y_2^+)	(y_1^+, y_2^-)	(y_1^+, y_2^+)
DC (r)	$<$	$<$	$<$	$<$	$\star$	$\star$	$\star$	$\star$
DC (l)	$>$	$>$	$>$	$>$	$\star$	$\star$	$\star$	$\star$
DC (u)	$\star$	$\star$	$\star$	$\star$	$<$	$<$	$<$	$<$
DC (d)	$\star$	$\star$	$\star$	$\star$	$>$	$>$	$>$	$>$
EC (r)	$<$	$<$	$=$	$<$	$\star$	$<,=$	$>,=$	$\star$
EC (l)	$>$	$=$	$>$	$>$	$\star$	$<,=$	$>,=$	$\star$
EC (u)	$\star$	$<,=$	$>,=$	$\star$	$<$	$<$	$=$	$<$
EC (d)	$\star$	$<,=$	$>,=$	$\star$	$>$	$=$	$>$	$>$
PO (r)	$<$	$<$	$>$	$<$	$\star$	$<$	$>$	$\star$
PO (l)	$>$	$<$	$>$	$>$	$\star$	$<$	$>$	$\star$
PO (r)	$=$	$<$	$>$	$<$	$<$	$<$	$>$	$\star$
PO (r)	$=$	$<$	$>$	$<$	$\star$	$<$	$>$	$>$
PO (r,l)	$>$	$<$	$>$	$<$	$<$	$<$	$>$	$\star$
PO (r,l)	$>$	$<$	$>$	$<$	$\star$	$<$	$>$	$>$
PO (l)	$>$	$<$	$>$	$=$	$<$	$<$	$>$	$\star$
PO (l)	$>$	$<$	$>$	$=$	$\star$	$<$	$>$	$>$
PO (u)	$\star$	$<$	$>$	$\star$	$<$	$<$	$>$	$<$
PO (d)	$\star$	$<$	$>$	$\star$	$>$	$<$	$>$	$>$
PO (u)	$<$	$<$	$>$	$\star$	$=$	$<$	$>$	$<$
PO (u)	$\star$	$<$	$>$	$>$	$=$	$<$	$>$	$<$
PO (u,d)	$<$	$<$	$>$	$\star$	$>$	$<$	$>$	$<$
PO (u,d)	$\star$	$<$	$>$	$>$	$>$	$<$	$>$	$<$
PO (d)	$<$	$<$	$>$	$\star$	$>$	$<$	$>$	$=$
PO (d)	$\star$	$<$	$>$	$>$	$>$	$<$	$>$	$=$
TPP (r)	$=$	$<$	$>$	$<$	$>,=$	$<$	$>$	$<,=$
TPP (l)	$>$	$<$	$>$	$=$	$>,=$	$<$	$>$	$<,=$
TPP (u)	$>,=$	$<$	$>$	$<,=$	$=$	$<$	$>$	$<$
TPP (d)	$>,=$	$<$	$>$	$<,=$	$>$	$<$	$>$	$=$
TPP (r,l,u,d)	$=$	$<$	$>$	$=$	$>$	$<$	$>$	$<$
TPP (r,l,u,d)	$>$	$<$	$>$	$<$	$=$	$<$	$>$	$=$
NTPP	$>$	$<$	$>$	$<$	$>$	$<$	$>$	$<$
EQ	$=$	$<$	$>$	$=$	$=$	$<$	$>$	$=$

Correspondence between IA and RCC8

The disadvantage of directly transforming the RCC8 relations into PA representations is that the total number of variables will be $4n$, where n is the number

of variables in the original RCC8 QCN, which would decrease the efficiency of constructing rectangular pseudo-solutions. Is it possible to reduce the total number of variables after transformation? In fact, we can first make use of the ORD-Horn IA relations $\mathcal{H}$. Similar to the concept of being PA representable, we say an RA relation is *Horn representable* if it is the product of two IA relations that are in $\mathcal{H}$.

Definition 6.3. Let R be a basic RCC8 relation. Suppose α and β are two IA relations in $\mathcal{H}$ and $\mathcal{N} = \{(x_1 \alpha x_2), (y_1 \beta y_2)\}$ is an IA QCN. We say that $\mathcal{N}$ is a *Horn representation* of R if $\alpha \otimes \beta$ is contained in R.

For a basic RCC8 relation R, we say that a set of its Horn representations *cover* R, if, for any two rectangles satisfying the RCC8 relation, the x and y projections of them also satisfy one of the Horn representations. Table 6.3 shows a choice of Horn representations that cover the basic RCC8 relations. Similar to the case of PA representations, we can also characterise these Horn Representations with relative positions "right", "left", "up", and "down". In Table 6.3, we also use "r", "l", "u", and "d" to specify the relative positions of each PA representation.

After transforming RCC8 relations into Horn representations, we obtain an IA QCN over $\mathcal{H}$, which can actually be divided into two IA QCNs over $\mathcal{H}$ for x- and y-projections respectively. Enforcing path consistency $\mathbf{PC}$ is able to decide the consistency of them. Moreover, Renz [98] showed the following result.

Proposition 6.4 ([98])**.** *Given a path consistent QCN $\mathcal{N}$ over $\mathcal{H}$, by applying the strategy in Algorithm 8, we are able to refine the $\mathcal{H}$ relations in $\mathcal{N}$ to PA representable IA relations while retaining the resulting QCN to be path consistent.*

It means that we can easily transform Horn representations into PA representations whenever needed, while retaining path consistency. With this

Table 6.3: Correspondence of basic RCC8 relations and Horn Representations.

	(x_1, x_2)	(y_1, y_2)
DC (r)	b	$\star$
DC (l)	bi	$\star$
DC (u)	$\star$	b
DC (d)	$\star$	bi
EC (r)	m	$\star \setminus \{\mathsf{b}, \mathsf{bi}\}$
EC (l)	mi	$\star \setminus \{\mathsf{b}, \mathsf{bi}\}$
EC (u)	$\star \setminus \{\mathsf{b}, \mathsf{bi}\}$	m
EC (d)	$\star \setminus \{\mathsf{b}, \mathsf{bi}\}$	mi
PO (r)	o	$\star \setminus \{\mathsf{b}, \mathsf{bi}, \mathsf{m}, \mathsf{mi}\}$
PO (l)	oi	$\star \setminus \{\mathsf{b}, \mathsf{bi}, \mathsf{m}, \mathsf{mi}\}$
PO (u)	$\star \setminus \{\mathsf{b}, \mathsf{bi}, \mathsf{m}, \mathsf{mi}\}$	o
PO (d)	$\star \setminus \{\mathsf{b}, \mathsf{bi}, \mathsf{m}, \mathsf{mi}\}$	oi
PO (r,l,u,d)	d, o, s, f	di, oi, si, fi
PO (r,l,u,d)	di, oi, si, fi	d, o, s, f
TPP (r)	s, eq	d, s, f
TPP (l)	f, eq	d, s, f
TPP (u)	d, s, f	s, eq
TPP (d)	d, s, f	f, eq
NTPP	d	d
EQ	eq	eq

Algorithm 8: RefH(R), an algorithm for refining a Horn IA relation into a PA representable IA relation.

Input: R, an IA relation in $\mathcal{H}$.
Output: R^*, an IA relation that is PA representable.

```
1  if R = {eq} then
2  |  R* ← R;
3  end
4  else
5  |  R' ← R ∩ {b, bi, o, oi, d, di};
6  |  if R' ≠ ∅ then
7  |  |  R* ← R';
8  |  end
9  |  else
10 |  |  R* ← R \ {eq};
11 |  end
12 end
13 return R*.
```

result, using Horn representations to construct pseudo-solutions would also be as convenient as using PA representations.

Algorithm

We will now present the algorithm for encoding any complete basic RCC8 QCNs, i.e. RCC8 scenarios, by rectangular pseudo-solutions. Here, we focus on dealing with a set of regions that implicitly induce an RCC8 scenario, while the idea can be easily adapted to the case of directly taking RCC8 basic constraints as input. The following discussion will be based on the Horn representations rather than the PA representations.

Recall that, when restricted to rectangles, each basic RCC8 relation is covered by a set of Horn representations as defined before. Therefore, we can incrementally build an IA QCN in a greedy fashion, from that we can construct a partial rectangular solution of the RCC8 QCN. Algorithm 9 and Algorithm 10 together give the detailed process.

Algorithm 9: EPS($\mathcal{N}$), an algorithm for constructing a pseudo-solution of an RCC8 scenario.

Input: $\mathcal{N}$, an RCC8 scenario with variables V.
Output: $\mathcal{L}$, a pseudo-solution of $\mathcal{N}$ (initially empty).

```
1 while N ≠ ∅ do
2 |   (S, N') ← CPS(N);
3 |   L.add(S);
4 |   N ← N';
5 end
```

In particular, on Line 2 of Algorithm 9, it repeatedly calls Algorithm 10 to construct partial solutions of the progressively weakened RCC8 QCNs. When all constraints have been removed from the original scenario $\mathcal{N}$, we know that any constraint from the original scenario is satisfied by a partial solution, and

Algorithm 10: CPS($\mathcal{N}$), an algorithm for constructing a rectangular partial solution.

Input: $\mathcal{N}$, an RCC8 QCN with variables V.
Output: A pair $(S, \mathcal{N}')$ where S is a rectangular partial solution of $\mathcal{N}$ and $\mathcal{N}'$ is a correspondingly weakened QCN.

```
1  I_s ← ∅;
2  V_s ← ∅;
3  for each variable v_0 ∈ V do
4  |   v_0.feasible ← true;
5  |   I'_s ← I_s;
6  |   for each variable v_i ∈ V_s do
7  |   |   if ∃Horn_{v_0 R v_i} path consistent with I'_s then
8  |   |   |   I'_s.add(Horn_{v_0 R v_i});
9  |   |   else
10 |   |   |   v_0.feasible ← false;
11 |   |   |   break;
12 |   |   end
13 |   end
14 |   if v_0.feasible then
15 |   |   V_s.add(v_0);
16 |   |   I_s ← I'_s;
17 |   end
18 end
19 S ← solution(I_s);
20 N' ← N \ N|_{V_s};
```

hence that $\mathcal{L}$ is a pseudo-solution of the original scenario. As for Algorithm 10, it repeats the following steps for each variable v_0 to construct a rectangular partial solution of a weakened RCC8 QCN $\mathcal{N}$.

- Line 6: Given that V_s is the set of variables that have already been considered in these steps, for each basic RCC8 constraint $(v_0 R v_i)$ between v_0 and $v_i \in V_s$, execute the following steps.
  - Line 7: choose a Horn representation $\mathsf{Horn}_{v_0 R v_i}$ for $(v_0 R v_i)$ that is path consistent with the current IA QCN I'_s being built (initially I'_s is

a copy of the IA QCN I_s);

– Line 9: if no such choice is possible, then move to the next variable.

- Line 16: If no inconsistencies have occurred, update the IA QCN I_s with I'_s, where all the chosen Horn representations have been added.

After this process, we obtain a path consistent IA QCN I_s corresponding to a subset V_s of variables in the RCC8 QCN $\mathcal{N}$. Note that I_s is a path consistent QCN over $\mathcal{H}$. By Proposition 6.4, we can transform I_s into a path consistent QCN I_s^* over PA representable IA relations, and then transform I_s^* into a path consistent PA QCN P_s. By using the idea of the algorithm by van Beek [121] based on topological sort, we can find a solution of P_s, from that a rectangular partial solution S of the RCC8 QCN $\mathcal{N}$ can be easily constructed. Then on Line 20, we weaken $\mathcal{N}$ to $\mathcal{N}'$ by removing all the constraints between the variables in V_s. Note that these constraints are satisfied by S. In fact, it is easy to see that the following conclusion holds.

Proposition 6.5. *Let S_i be the partial solution that is constructed in the i-th iteration of the while loop in Algorithm 9 and let V_i be the corresponding set of variables. It holds that any constraint (uRv) from the original RCC8 QCN $\mathcal{N}$ is satisfied by the partial solution S_{i_0}, where i_0 is the smallest index for which $u, v \in V_{i_0}$.*

After each iteration of the while loop in Algorithm 9, at least one constraint is removed. Therefore the algorithm will terminate after at most $O(n^2)$ iterations, where $|V| = n$. In practice, a larger number of constraints would typically be removed in each iteration, and thus we can expect the required number of iterations actually to be much smaller. The number of operations taken by Algorithm 10 is polynomial in the number of variables, because we

add IA constraints corresponding to at most $O(n^2)$ RCC constraints. By incrementally checking the path consistency, the total number of operations by Algorithm 10 is bounded by $O(n^5)$.

6.2.2 Clustering

When the number of variables becomes very large, it would not be feasible to directly apply our approach to the whole RCC8 scenario. To address this, we propose to cluster the variables and apply the approach to the RCC8 scenario for each of these clusters. Note that this is different from the use of clusters by Fogliaroni [43] discussed in the previous chapter. The idea of Fogliaroni was to remove constraints between variables in different clusters by using the so-called clustering relations. In contrast, we use clusters to decompose the large scenario into smaller scenarios, so that Algorithms 9 and 10 can be efficiently applied to compactly encode the constraints between the variables in each cluster.

When we start from a set of regions rather than the explicit RCC8 scenario, we use the idea of Quadtree [42] to obtain a suitable clustering. In particular, the space of the regions is first split into $N \times N$ grid cells of equal size, where N is called the initial grid size. A region is assigned to a grid cell if it has a common point with that cell. If the number of regions assigned to a single grid cell exceeds a given limit K, then we split that cell into four cells of equal size, and repeat the procedure until either all grid cells are assigned with less than K regions or the maximum number of splits M has been arrived at. Note that in practice there might be common points that belong to more than K of the regions, in which case the resulting clusters will always contain more than K regions. Furthermore, if two regions are connected, i.e. they have a common point, then there is always at least one cluster to which they both belong.

We then use Algorithms 9 and 10 to generate a sequence of rectangular pseudo-solutions for the RCC8 scenarios of these clusters. Moreover, assuming that the clusters are ordered in some way, we will only consider the RCC8 constraint (uRv) in the first cluster that contains both u and v, i.e. we remove this constraint from the RCC8 QCNs of all succeeding clusters.

6.2.3 Answering Queries

To retrieve the RCC8 relation R_{ij} between v_i and v_j, note that the constraint $(v_i R_{ij} v_j)$ is only considered in the first cluster that contains both v_i and v_j. Then we first need to find out the pseudo-solution corresponding to this cluster. After determining the pseudo-solution, we need to find the first partial solution in it that covers both v_i and v_j. If there are no clusters that contain both variables, it means that the RCC8 relation between them is **DC**. Algorithm 11 shows the procedure.

Algorithm 11: QPS(v_i, v_j), an algorithm for retrieving an RCC8 relation from a pseudo-solution.

Input: (v_i, v_j), a pair of variables.
Output: R, the RCC8 relation between them.

```
1 L_c ← FindPseudoSol(v_i, v_j);
2 if L_c does not exist then
3 |   return DC;
4 end
5 S ← FindPartialSol(v_i, v_j);
6 R ← CalculateRelation(S(v_i), S(v_j));
7 return R.
```

To allow for efficient query answering, we store the information about each variable v_i as follows.

- An array with the indices of the pseudo solutions corresponding to the

clusters that contain v_i, sorted in ascending order. We call this array the *pseudo-solution array* of v_i.

- For an entry corresponding to a cluster $\mathcal{C}_k$in the pseudo solution array of v_i, we associate an array with the indices of the partial solutions (for $\mathcal{C}_k$), that involve v_i, sorted in ascending order. We call the array a *partial solution array* of v_i w.r.t. the cluster $\mathcal{C}_k$.

The total storage size is then proportional to the total number of rectangles. For a dataset, the number of variables is fixed, and hence the total storage size will be determined by the average number of rectangles constructed for a variable.

We now analyse in more detail about using the proposed representation to answer queries efficiently, i.e. to execute FindPseudoSol(v_i, v_j) and FindPartialSol(v_i, v_j).

For FindPseudoSol(v_i, v_j), the relevant pseudo-solution for a given pair of regions (v_i, v_j) can be found, by using the corresponding pseudo-solution arrays A_i and A_j, in $O(|A_i| + |A_j|)$ steps. Let us write $|A_i| = a_i$. Suppose each ordered pair (v_i, v_j) $(i \neq j)$ is queried with equal probability. Then the average number of comparisons is given by:

$$O(\frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{i \neq j} (a_i + a_j)) = O(\frac{1}{n} \sum_{i} a_i) = O(\frac{1}{n} \sum_{k} |\mathcal{C}_k|),$$

where n is the number of pseudo-solutions (and also the number of clusters) and $|\mathcal{C}_k|$ is the number of regions in cluster $\mathcal{C}_k$. In other words, the average number of comparisons is proportional to the total cardinality of the clusters.

After determining the corresponding pseudo-solution, we need to find out the first partial solution in the pseudo-solution that specifies a rectangle for each of the two variables, i.e. to execute FindPartialSol(v_i, v_j). Let b_i be the

number of partial solutions in the pseudo-solution corresponding to $\mathcal{C}_k$, which in total specify b_i rectangles for v_i. If each pair of (v_i, v_j) is queried with equal probability, then, similar to the case of finding the relevant pseudo-solution, we can find the first partial solution which specifies rectangles to the two variables v_i and v_j by $O(\frac{1}{t}\sum_i b_i)$ comparisons on average, where t is the number of variables in the considered cluster. In other words, the average number of comparisons for determining the relevant partial solutions is proportional to the total number of specified rectangles in the considered pseudo-solution. When the cluster is fixed, the number of variables will be fixed too, and thus the average number of comparisons is determined by the average number of rectangles constructed for each variable in that cluster.

As we can see from the above analysis, to reduce the storage size and to answer queries efficiently, we should (i) reduce the total cardinality of the clusters and (ii) reduce the average number of rectangles that is constructed for a variable in a pseudo-solution. Later we will discuss how we could tackle these in implementation.

6.2.4 The Non-DC Method

In the previous chapter, we have known that for RCC8 information, the method Non-**DC** always has a more compact representation than using MC, MD, or the grid/R-tree clustering indexes. Therefore, in the experiments, we will use Non-**DC** as our main baseline. Given a set of n variables, let $N^c = \{v_i R_{ij} v_j : R_{ij} \neq \textbf{DC}, 1 \leq i < j \leq n\}$ be the set of non-**DC** constraints, i.e. the ones stored by Non-**DC**. Note that on average, each region intersects with $2|N^c|/n$ regions in the dataset. In the following, we will refer to $|N^c|/n$ as the *Intersection Measurement Index* (IMI) for the dataset, and use it as the measure of the performance of Non-**DC** on a dataset.

In terms of actual storage, we first note that both methods depend on various implementation factors such as data structures. Here we give a rough analysis. For Non-**DC**, to store a relation for two regions, assume that we use the 32-bit int data type to store the ID of a region and 3 bits to store a relation. Then Non-**DC** needs $(32 \times 2 + 3)|N^c| = (16 \times 4 + 3)|N^c|$ bits in total, where $|N^c|$ is the number of non-**DC** relations. For the pseudo-solution representation, to store a rectangle, assume that we use the 32-bit int data type. Then in total it needs $32 \times 4 \times k$ bits, where k is the number of rectangles. In addition, to store the indices of pseudo-solutions and partial solutions, assume that we also use the 32-bit int data type. Note that the number of pseudo-solutions and the number of partial solutions are both smaller than the number of rectangles. Then we need additional $32 \times 2 \times k$ bits at most. Moreover, because for a given cluster and for a partial solution of the cluster, the number of rectangles is small (smaller than the number of variables in the cluster), we can instead use the 16-bit short int data type for the endpoints of rectangles, which reduces the total storage size for rectangles to be $16 \times 4 \times k$. Also, for each cluster, the number of partial solutions is usually small, and we can again use the 16-bit short int data type. Thus, the total storage size of the pseudo-solution representation will be $16 \times 7 \times k$. It is worth noting that in some applications (e.g. [111]), the relation from a to b and the relation from b to a are both stored for efficient retrieval of both relations, and in this case Non-**DC** needs double of the previous storage size, i.e. $(16 \times 8 + 3)|N^c|$ bits, while the pseudo-solution representation has already had the support for retrieving both relations and the storage size will remain $16 \times 7 \times k$ bits.

In the experiments, in order to see the essence of the performance of the pseudo-solution representation, we will use k/n, the average number of rectangles, as the measure to compare with the IMI value $|N^c|/n$, i.e. the measure

of Non-**DC**.

Besides the storage size, we note again that Non-**DC**, as well as the two derivatives MC and MM of MA, is based on some kind of close world assumption, and will return an incorrect relation if the actual relation was lost by accident. On the other hand, the pseudo-solution representation can avoid this problem. In fact, the only **DC** relations that are not encoded by rectangles are between variables that are never contained in the same cluster. Therefore, we can assign each variable an additional rectangle where these rectangles are pairwise disjoint, and then all of these **DC** relations will be corrected encoded.

6.3 Implementation

In this section, we discuss some implementation details and improvements of the algorithm, which affect the overall performance.

For clustering, we restrict K, the (soft) limit of the number of regions in each cluster, to be 500. In practice, this limit should be chosen as large as possible to reduce the total cardinality of the clusters, while keeping the number of the variables in the clusters feasible for the algorithm. For each dataset, we first test different values of the initial split size N by clustering the variables with them, to see which value gives smaller total cardinality of the clusters. Note that $N = 1$ is not always optimal. In fact, for some datasets used in our experiments, the optimal value of N would be larger values such as 11. For each of the clusters obtained by clustering with the optimal value of N, we generate a pseudo-solution containing a sequence of rectangular partial solutions.

For the algorithm of constructing a pseudo-solution, the crucial part is to generate partial solutions, i.e. CPS (Algorithm 10). In CPS, first note that

some variables in V_s might not have constraints with any of the other variables in V_s, because these constraints might have already been satisfied by earlier partial solutions and removed from the QCN. In such a case, it is not necessary to include a rectangle for each of these variables in the new partial solution. Therefore, we can generate a partial solution for the current QCN w.r.t. the variables in V_s without such ones. This will reduce the total number of rectangles in the pseudo-solution without affecting the correctness of the algorithm. The following discussions will always apply such operation when generating a partial solution.

6.3.1 Optimisations

Besides, there are three critical steps in Algorithm 10 that affect the number of rectangles in the resulting pseudo-solutions:

1. Line 3: How to choose v_0?
2. Line 6: How to choose v_i?
3. Line 7: How to choose a Horn representation of $(v_0 R v_i)$?

In a naive implementation of the algorithm, which we will refer to as Naive, we simply consider a random ordering of the variables, and choose the first Horn representation according to the static ordering as in Table 6.3.

However, the order to choose variables as v_0 and the order to choose variables in V_s as v_i will affect the results significantly. The major reason would be that, for a set of variables, the constraints associated with them will affect the likelihood of finding a rectangular partial solution for them. For example, suppose v_0 has **EC** relations with variables $\{v_1, \ldots, v_5\}$ that are pairwise **DC**. If we first add $\{v_1, \ldots, v_5\}$ into V_s, it might be impossible to find rectangles

for these five variables such that all the **EC** constraints are satisfied. Note that the rectangle of v_0 should touch the rectangles of the variables $\{v_1, \ldots, v_5\}$. Then, after choosing the Horn representations corresponding to some specific relative positions of the disjoint rectangles of $\{v_1, \ldots, v_5\}$, it will be impossible to find a rectangle for v_0 such that the rectangle is touching each rectangle of $\{v_1, \ldots, v_5\}$.

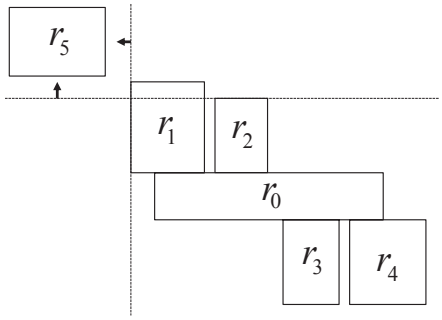


Figure 6.4: Illustration of the dilemma of the order of choosing Horn representations.

For example, let $r_1, \ldots, r_5$ be the rectangles for $v_1, \ldots, v_4$. Suppose we have chosen that

1. r_2 is to the right of r_1;
2. r_3 is to the right of r_2 and to the downside of r_1;
3. r_4 is to the right of r_3 and to the downside of r_1 and r_2;
4. r_5 is to the left of r_1 and to the upside of r_2.

Because of Choices 1, 2, 3 and 4, r_1 can only touch the upside edge of r_0. From Choices 2 and 3, r_2 can also only touch the upside edge of r_0. However, under such restrictions, we cannot find r_0 for v_0 such that r_5 also touches it. Figure 6.4 illustrates this dilemma. For the order of choosing v_i, it would have

a similar circumstance. This suggests that we should order the variables based on the kind of relations in which they are involved.

The optimisation Label

Based on the above intuition, we propose the following improvement. For every variable, we count the number of times it is involved in each of the basic RCC8 relations. Using n_R^v to denote such a number, we then order the variables as follows. For variable v, let the vector

$$(n_{R_1}^v, n_{R_2}^v, n_{R_3}^v, n_{R_4}^v, n_{R_5}^v)$$

contain the number of R_1, ..., R_5 relations for v. Here, we consider $(R_1, \ldots, R_5)$ corresponds to a permutation of the five relations **DC**, **EC**, **PO**, **TPP(i)**, and **NTPP(i)**, where **TPP(i)** means the relation is either **TPP** or **TPP**$^{-1}$ and similarly **NTPP(i)** means the relation is either **NTPP** or **NTPP**$^{-1}$. Here we do not distinguish **TPP** and **TPP**$^{-1}$ (and **NTPP** and **NTPP**$^{-1}$) because we think they are of the same importance for they are the inverse of the other.

To order variables, we use the lexicographic order between these vectors. In other words, we first order the variables according to $n_{R_1}^v$. To break ties, we first consider $n_{R_2}^v$, etc. We will refer to this improvement as Label.

We generated several synthetic datasets to test which permutation of the five of relations has the best performance. In particular, for each IMI value in $\{13, 20, 28, 37, 46\}$, we generated 10 datasets of 100 simple regions (i.e. convex polygons), and for each IMI value in $\{15, 23, 28, 35, 42\}$, we generated 10 datasets of consistent RCC8 scenarios of 100 variables. To generate a convex polygon, we sample some elements from a set of points with integer coordinates, say $\{(i, j) : 0 \leq i, j \leq d\}$, and calculate the convex hull of the se-

lected points. To sample the points, we first randomly select a "feed" point, which induces a Gaussian distribution on the integer points around it, and then we randomly sample the other points from this Gaussian. To generate consistent RCC8 scenarios, we exploit the Barabási-Albert (BA) model used in Section 3.6.3 in Chapter 3. For each IMI value, we extracted 10 RCC8 scenarios by using the technique introduced in [114] from the scale-free structured QCNs generated by the BA model (with the preferential attachment of 2). More specifically, for the scenarios with the IMI value 15, we have set the percentages of the relations **DC**, **EC**, **PO**, **TPP(i)**, **NTPP(i)** to be $76\%, 10\%, 10\%, 2\%, 2\%$, and for the other IMI values, we reduce the percentage of **DC** relations accordingly and increase the percentages of the other relations by amounts proportionally to their percentages. Such percentages of relations were chosen to mimic the QCNs representing the RCC8 relations between regions.

For these datasets, we count the average number of rectangles in the pseudo-solution generated by each permutation of the five types of constraints. Due to the large number of permutations, in Table 6.4 we only report the minimum average number of rectangles generated for the 10 datasets of simple regions (the second row of the table), and that for the 10 datasets of BA scenarios (the fourth row of the table).

Table 6.4: Minimum average numbers of rectangles generated by Label with different permutations for synthetic datasets.

IMI (Simple Regions)	13	20	28	37	46
Min. Avg. #Rect	3.30	3.70	3.96	3.76	3.32
IMI (BA Scenario)	15	23	28	35	42
Min. Avg. #Rect	10.51	12.61	13.64	14.99	15.96

The average numbers of rectangles generated by using the permutation $(\textbf{NTPP(i)}, \textbf{TPP(i)}, \textbf{PO}, \textbf{EC}, \textbf{DC})$ are shown in Table 6.5. In fact, by checking the average number of rectangles generated by each permutation of the five types

Table 6.5: Average numbers of rectangles generated by Label with the permutation (**NTPP(i)**, **TPP(i)**, **PO**, **EC**, **DC**) for synthetic datasets.

Simple IMI	13	20	28	37	46
Simple	3.30	3.70	4.09	3.76	3.55
BA IMI	15	23	28	35	42
BA	10.90	12.81	13.98	15.02	16.11

of constraints, we found that this permutation leads to the smallest average residual value and a moderately good performance, compared to the minimum values in Table 6.4. One can see, by comparing these tables, that there is only a small difference between the values of the chosen permutation and the minimum values of all considered permutations.

Note that the containment relations are ordered before the others. One possible reason could be that the relative position of one rectangle involved in a containment relation has a great influence on the relative position of the other rectangle. On the other hand, it is usually easy to find a rectangle that satisfies a given **DC** constraint. In the following, Label will use $(n^v_{\textbf{NTPP(i)}}, n^v_{\textbf{TPP(i)}}, n^v_{\textbf{PO}}, n^v_{\textbf{EC}}, n^v_{\textbf{DC}})$ to order the variables.

The optimisation Weight

We also consider the following alternative. With each variable v we associate a score s_v defined as:

$$s_v = w_1 n^v_{R_1} + w_2 n^v_{R_2} + w_3 n^v_{R_3} + w_4 n^v_{R_4} + w_5 n^v_{R_5},$$

where $(R_1, \ldots, R_5)$ corresponds to a permutation of the five relations **DC**, **EC**, **PO**, **TPP(i)**, and **NTPP(i)**. We can then order the variables according to these scores. As w_i can be seen as the weight for R_i, this improvement will be referred to as Weight.

Moreover, based on the result of Label, we assume that $(R_1, \ldots, R_5) = (\textbf{NTPP(i)}, \textbf{TPP(i)}, \textbf{PO}, \textbf{EC}, \textbf{DC})$ and $w_1 > w_2 > w_3 > w_4 > w_5$. We should note that it is not possible to test all the values of the weights w_i, and thus we will only choose the best one from a manually selected candidate set of weights. To generate a candidate set of weights, we first let the initial values to be $(4, 3, 2, 1, 0)$. Here w_5 is set to 0 because we want to prevent the number of **DC** relations from affecting the selection of v according to the other relations. Then we generate several other candidate values by increasing the values for w_1, w_2, and w_3 up to 15, while maintaining the ordering $w_1 > w_2 > w_3$. In Table 6.6 we report the minimum average number of rectangles generated by Weight with these values for the 10 datasets of simple regions (the second row of the table), and that for the 10 datasets of BA scenarios (the fourth row).

Table 6.6: Minimum average numbers of rectangles generated by Weight with different values for w_i $(i = 1, \ldots, 5)$ on synthetic datasets.

IMI (Simple Regions)	13	20	28	37	46
Min. Avg. #Rect	3.24	3.70	3.81	3.67	3.48
IMI (BA Scenarios)	15	23	28	35	42
Min. Avg. #Rect	10.46	12.29	13.34	14.41	15.39

Table 6.7 shows the average number of rectangles generated by Weight with the values $(10, 5, 2, 1, 0)$. In fact, among the candidate values, we found that the result with the values $(10, 5, 2, 1, 0)$ shown in Table 6.7 has the smallest residual number and a moderately good performance, compared to the minimums in Table 6.6. Again, the average numbers of generated rectangles of the chosen weights are close to the minimum numbers for all the considered weights. In the following, we will use $s_v = 10n^v_{\textbf{NTPP(i)}} + 5n^v_{\textbf{TPP(i)}} + 2n^v_{\textbf{PO}} + 1n^v_{\textbf{EC}}$ as the score of a variable v in Weight.

Table 6.7: Average numbers of rectangles generated by Weight with the values $(10, 5, 2, 1, 0)$ on synthetic datasets.

IMI (Simple Regions)	13	20	28	37	46
Min. Avg. #Rect	3.48	3.82	4.07	3.76	3.67
IMI (BA Scenarios)	15	23	28	35	42
Min. Avg. #Rect	10.98	12.68	13.62	14.65	15.67

The optimisation Type

Finally, we consider the third critical point that affects the performance of the algorithm, i.e. how to choose a Horn representation for RCC8 relations that do not correspond to a unique Horn representation. Assume we have a set of regions rather than the explicit RCC8 relations. We use a heuristic way to choose Horn representations. In particular, note that a Horn representation of a basic RCC8 relation corresponds to some configurations of rectangles, and these rectangles have relative positions such as *left*, *right*, *up*, and *down* (cf. Table 6.3). After finding out the relative position of the MBRs, we check if the MBRs are in the same relation as the regions. If this is the case, we will consider first the Horn representations corresponding to the same relative position of the MBRs. Otherwise, we look at the relative position of the centre points of the MBRs. The Horn representations that have the same relative position (e.g. *right*) as the centre points will be considered first. We will refer to this improvement as Type.

Usually, the relative position of the centre points of the MBRs is vague, e.g., one lies at the right up corner of the other. Therefore, we introduce the function $1/(1 + e^{k\alpha})$ to assign probabilities to choosing one of the relative position, where k controls the steepness of the transition of probabilities from 0 to 1. We tested different values of k from 1 to 20 on the synthetic datasets consisting of convex polygons. It turns out that the results with $k = 2$, shown

in the third row of Table 6.8, have the most stable and the smallest residuals compared to the minimum numbers, as shown in the second row of Table 6.8. In the following, we will set $k = 2$ in Type.

Table 6.8: Minimum average numbers of rectangles generated by Type with different values of k and average numbers of rectangles generated by Type with $k = 2$ for synthetic datasets.

IMI (Simple Regions)	13	20	28	37	46
Min. Avg. #Rect	3.41	4.16	4.41	5.26	5.45
Avg. #Rect for $k = 2$	3.72	4.42	5.00	5.41	5.89

6.3.2 Comparison of Optimisations

To compare the performance of the different variants of our algorithm, we first use the previous synthetic datasets. In the above, we have already obtained the results of Label, Weight, and Type on the datasets of convex polygons, as well as the results of Label and Weight on the RCC8 scenarios extracted from the BA model. Here, we also include Naive and the other two variants that combine Label or Weight and Type. For all the tests in this subsection, we did not cluster the regions or the variables.

Table 6.9 summaries the average number of rectangles for each variant of our algorithm on the datasets of convex polygons. We can see from the table that Weight and Label perform similarly, and substantially outperform Naive. In most cases, Type alone is not as effective as Weight and Label, while the combination of Label/Weight and Type usually leads to the best performance.

For the RCC8 scenarios extracted from the BA model, the performance of each of these variants decreases (see Table 6.10). This is mainly because it is already hard to find 2D simple geometric solutions of these constraints. The method Naive results in a significant larger average number of rectangles than

Table 6.9: Comparison of the average number of rectangles needed for different implementations on the datasets of 100 convex polygons.

IMI	Naive	Weight	Label	Type	Weight + Type	Label+Type
13	13.70	3.48	3.30	3.72	3.08	2.76
20	14.05	3.82	3.70	4.42	3.12	3.14
28	12.62	4.07	4.09	5.00	3.24	3.31
37	11.71	3.76	3.76	5.41	3.36	3.53
46	10.86	3.67	3.55	5.89	3.02	3.40

Table 6.10: Comparison of the average number of rectangles generated by different implementations for the RCC8 scenarios of 100 variables extracted from the BA model.

IMI	Naive	Weight	Label
15	31.54	10.98	10.90
23	36.66	12.68	12.81
28	38.42	13.62	13.98
35	39.35	14.65	15.02
42	40.06	15.67	16.11

the IMI value for each of these datasets. On the other hand, both Weight and Label still notably improve Naive and have much smaller numbers of rectangles than the IMI values. Note that variants involving Type are not applicable here since these datasets have no geometric information.

We also compared the variants on Real-2 consisting of five small real-world datasets used in the previous chapter. These datasets have cardinalities about 600, and IMI values of 12, 30, 46, 52, and 74. Table 6.11 shows the average number of rectangles for each variants of our algorithm. Similar to the previous results, Label and Weight still have good performance. All the others improve Naive, while Label + Type and Weight + Type did not improve Label and Weight. This is probably because that in this case many regions consist of multiple connected components, where the MBRs or centre points of MBRs seems less effective to reflect the correct relative positions and hence Type

Table 6.11: Comparison on small real-world datasets of the average number of rectangles needed for different implementations.

#var	IMI	Naive	Weight	Label	Type	Weight + Type	Label+Type
600	12	61.15	6.67	6.61	16.12	8.07	8.16
610	30	52.71	10.14	10.21	22.56	13.17	12.90
605	46	44.87	12.33	12.26	27.81	15.15	14.91
611	52	90.47	16.65	16.30	35.99	20.47	19.99
604	74	70.74	18.78	20.46	44.55	24.96	26.10

would be likely to order PA representations in a wrong way. Note that all these implementations except Naive outperform the Non-**DC** method in all cases, in the sense that the average number of required rectangles is smaller than the average number of relations that need to be stored by the Non-**DC** method, with the latter being equal to the IMI value. In the following experiments, we will focus on the optimisations Label and Weight.

6.4 Empirical Evaluation

Although the idea of the pseudo-solution approach could handle either an RCC8 scenario directly or a scenario that is implicitly induced by a set of regions, we will mainly consider the latter here, since we are mostly interested in the cases where the number of regions is too large to be explicitly represented as an RCC8 scenario.

To test the performance of our method on large real-world datasets, we have used several datasets in the previous chapter, including four datasets (HEU, HMS, SEU, and SMS) about species distribution and habitat from the European Environment Agency[1] (EEA), a dataset (CS) with county subdivisions[2] of the USA, a dataset (SC) of school catchment areas[3] in the USA, and

[1] http://www.eea.europa.eu/
[2] http://www.census.gov/
[3] http://nces.ed.gov/surveys/sdds/sabs/

the combination of the last two (CS+SC). The four datasets from EEA contain $5{,}322$, $6{,}258$, $10{,}061$ and $11{,}613$ regions (after removing duplicates), with an IMI of respectively 63.92, 61.83, 121.54, and 119.91. The dataset CS contains $36{,}702$ regions, with an IMI of 3.10. The dataset SC contains $65{,}192$ regions (after removing duplicates), with an IMI of 9.36. The dataset CS+SC contains $101{,}894$ regions with an IMI of 7.11. For the datasets HEU, HMS, SEU, and SMS, the percentages of **EC**, **PO**, **TPP(i)**, **NTPP(i)** among the non-**DC** relations are about $68.4\%, 25.7\%, 5.8\%, 0.1\%$ respectively. For SC, the percentages are about $42.8\%, 41.2\%, 10.2\%, 5.8\%$ respectively. For CS+SC, the percentages are about $51.8\%, 34.7\%, 8.6\%, 4.9\%$ respectively. The dataset CS only contains **DC** and **EC** relations, where the percentage of the **EC** relations is about 0.2%. As the number of variables in these datasets is very large, we will cluster the variables by setting the limit size of a cluster as 500.

6.4.1 Comparison With Baseline Methods

In the experiments, we use the Non-**DC** method as our main baseline, since for RCC8 information it leads to more compact representations than the grid or the R-tree clustering index and the MBR approaches MA and MD, according to the discussion in the previous chapter. As we have analysed in Section 6.2.4, we will measure the storage size of Non-**DC** by the IMI value for each dataset (i.e. the number of non-DC constraints averaged over the number of regions). The storage size of our algorithm will be measured by the average number of rectangles in the pseudo-solution representation.

In addition to comparing our results against the Non-**DC** method, we also present the results of the two derivatives of the MBR-based approach, i.e. MC and MD. Note that in general, the Non-**DC** method is not guaranteed to outperform these two methods. Both of them need to store the MBRs of regions

besides the constraints, and MD also needs to store the MBRs of connected components of the regions. Therefore, we measure the storage size of both methods by calculating the average number of stored MBRs and relations for each variable.

For the prime subnetwork method discussed in Chapter 4, currently it does not provide efficient enough strategy for retrieving the relation between two given variables, but resolves to qualitative reasoning on the whole network, which can be very inefficient when the number of variables becomes large ($O(n^3)$ worst case time complexity). As mentioned before, even for prime subnetworks with about 600 variables for the Real-2 datasets, it takes 10^7ns to calculate the relation between two variables whose constraint has been characterised as redundant and removed in the prime subnetwork. Therefore, we do not consider it as a comparable representation, and did not include it as a baseline here.

Table 6.12 shows the results on the real-world datasets. We can see that for the datasets HEU, HMS, SEU, and SMS, where the IMI values are relatively large, the number of non-**DC** constraints for each region is about 2.5 to 3.4 times larger than the number of rectangles generated by Label and Weight. For the datasets with very small IMI value but large number of regions, the results of Label and Weight are still comparable with and sometimes even better than the other approaches. This is what we may expect in cases where the IMI is small. By contrast, when the IMI becomes larger the differences will become more pronounced. For example, for each of the SEU and SMS datasets, the total number of non-**DC** constraints is about $1{,}400{,}000$, while the total number of rectangles generated by Label or by Weight is only about $440{,}000$.

Note that the average number of rectangles generated by Label and Weight becomes larger when the IMI increases. To check the connection between the

Table 6.12: Comparison of storage sizes of Label, Non-**DC**, MC, and MD for large real-world datasets.

	HEU	HMS	SEU	SMS	CS	SC	CS+SC
Label	23.37	24.68	35.63	37.96	3.31	6.62	5.40
Weight	24.38	25.60	36.97	38.57	3.45	6.32	5.29
Non-**DC**	63.92	61.83	121.54	119.91	3.10	9.36	7.11
MC	118.10	113.93	205.82	205.27	4.37	9.03	7.35
MD	74.22	69.94	127.32	122.79	4.50	9.04	7.41

growth rate of the number of generated rectangles and that of the IMI, we performed an additional experiment on synthetic data. For each of the IMI values from 11 to 223 with a step size around 15, we generated 10 datasets of 500 regions. Each of the regions is a set of disconnected polygons and the polygons can be concave. The number of disconnected polygons for each region is randomly chosen between 1 and 10. A concave polygon is generated as the previous process of generating convex polygons, by replacing the convex hull with the concave hull of points. When clustering, we set the limit size of a cluster to be 300. Note that if instead we set the limit size of a cluster to be 200, some clusters may have to be split many times, which will make clustering perform poorly and even fail. In this experiment we also included the results without clustering as a comparison.

Figure 6.5 illustrates the number of rectangles generated by Label, in relation to the IMI of the dataset. Note that each data point is averaged over 10 datasets. The result of Weight is almost the same as Label and for clarity we have removed it from the figure. From the result, we can see that both the growth rates of Label with clustering (Label-Cluster in the figure) and without clustering are similar and are much smaller than that of the IMI. By comparing the results with and without clustering, we also observed that the side effect of clustering becomes more noticeable when the IMI increases, as the difference

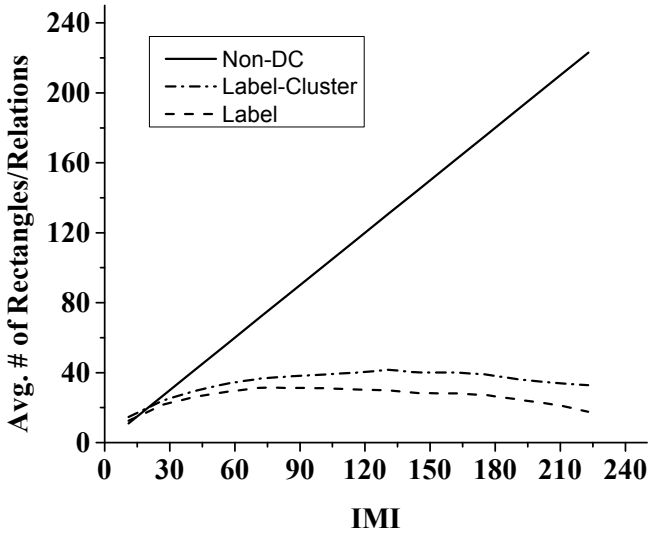
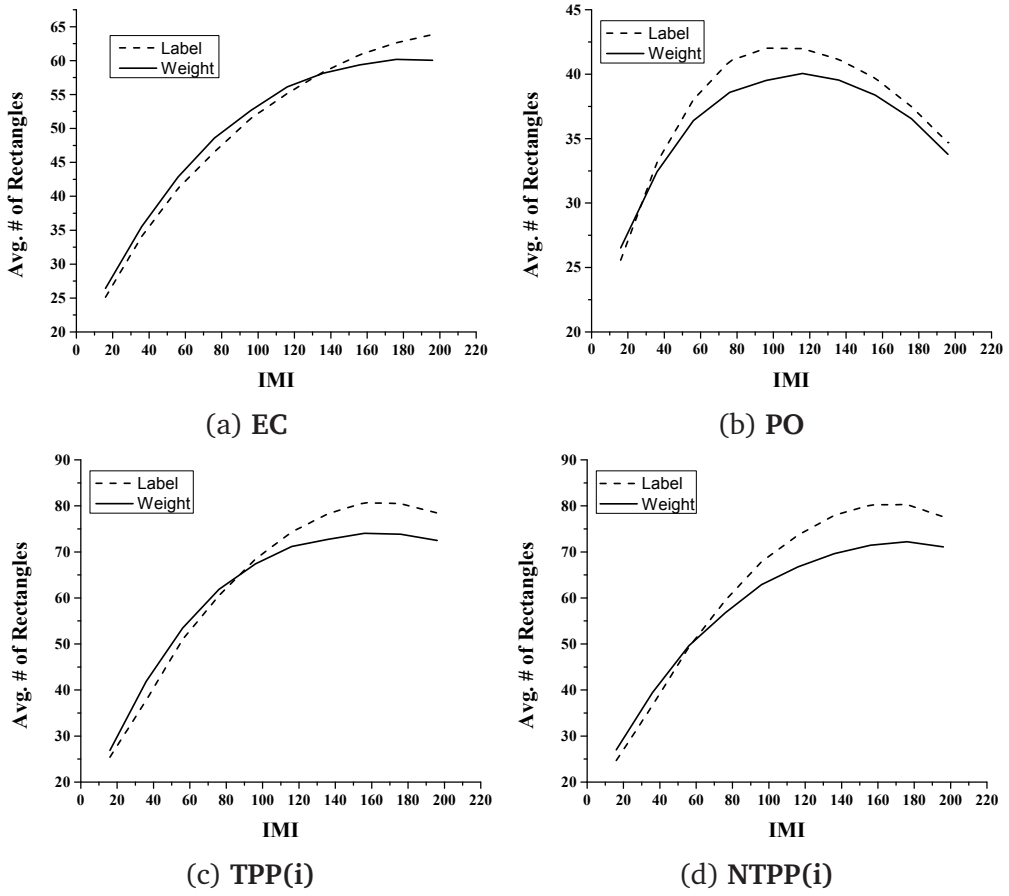


Figure 6.5: Illustration of the growth rate of Label with and without clustering when IMI increases.

in the number of generated rectangles becomes larger. The reason for the side effect of clustering is that, when the IMI increases, a larger number of variables will be included in several clusters and hence some extra rectangles will be generated for these variables.

In Figure 6.5, we also observed another interesting phenomenon, i.e. when IMI grows to be relatively large, the number of rectangles generated by Label (and Weight) without clustering on the contrary becomes smaller. In that case, the QCNs of the datasets contain more **PO** relations than the other relations except **DC**, i.e. **PO** relations becomes dominant. To have a closer examination of the effect of different types of relations in the QCN, we conducted experiments on several randomly generated RCC8 scenarios, which are extracted from the QCNs generated by the BA model. In particular, for each of the IMI values from 16 to 196, with a step size of 20, we extracted 10 complete basic RCC8 QCNs of 500 variables. When the IMI value increases, the percentage of the relation R also increases, starting from p_0 with a step size of about 0.08, where R can be

EC, **PO**, **TPP(i)**, and **NTPP(i)**, and p_0 is 10% for **EC** and **PO** and 2% for each of **NTPP(i)** and **TPP(i)**. The results are shown in Figure 6.6.



(a) **EC** (b) **PO**

(c) **TPP(i)** (d) **NTPP(i)**

Figure 6.6: The number of rectangles generated by Label and Weight on datasets with different dominant relations.

From the results, we can see that, for the case of where the number of **PO** relations becomes dominant, the number of rectangles generated by Label and Weight decreases visibly when the IMI grows close to the maximum value. For the case of **TPP(i)** and **NTPP(i)**, when the IMI approaches the maximum value, the number of rectangles also decreases, but only for a small amount. For the case of **EC**, the number of rectangles keeps increasing, for which we should note that there might not exist convex shaped solutions to the QCNs. Also,

for the case where **PO** dominates the other relations, the number of generated rectangles is smaller than that for the cases where **EC**, **TPP(i)**, or **NTPP(i)** dominates. The reason for this phenomenon is probably that the **PO** relations are easier to be satisfied by rectangles than the other types of relations. In fact, for instance, when all the relations are **EC**, the rectangular pseudo solutions will contain $O(n^2)$ number of rectangles. Moreover, **TPP(i)** and **NTPP(i)** relations seem difficult to be captured by rectangles, particularly when they are mixed with other relations. This might due to that when one rectangle contains another, these two rectangles are likely to have the same relations with the other rectangles, while usually it is not the case in the randomly generated QCNs. Nevertheless, the number of rectangles is still smaller than the number of non-**DC** constraints in most of the datasets tested, especially when the IMI becomes large. In addition, we expect that, in large real-world datasets of regions, these "degenerated" cases of QCNs are rare, since we have shown that our approach works well for such datasets in previous experiments.

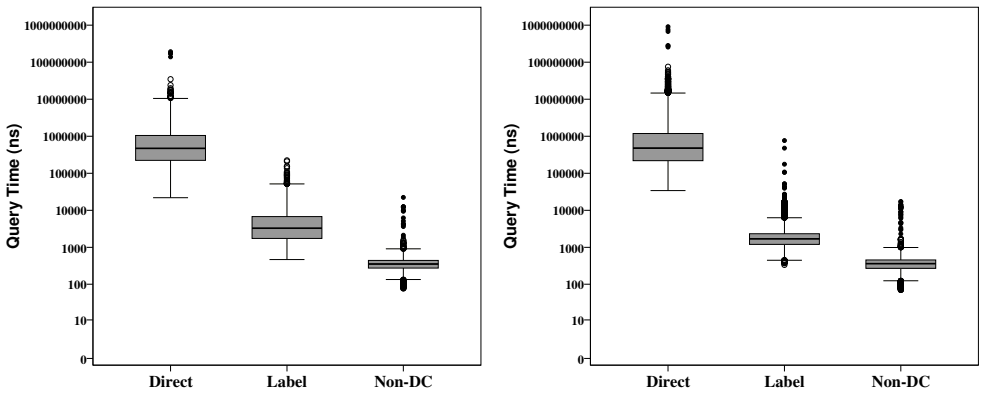
6.4.2 Answering Queries

Next we consider the computation time that is needed to determine the RCC8 relation between two given variables. This corresponds to the evaluation of queries such as "does species A live in an area where species B is also present" or "is neighbourhood X is the catchment area of school Y". This kind of queries are common in applications such as GIS and are fundamental ones. More complex queries can be answered by integrating our approach with other techniques, such as using the pseudo-solution representation as a compact backend representation of the R-tree method in [95]. In the following, we will compare the performance of three methods: (i) determining the relation by comparing the geometric representation of the boundaries of the regions by

using JTS[4] (Direct), (ii) using the pseudo-solution produced by the method Label[5], and (iii) using an RCC8 QCN without **DC** constraints (Non-**DC**). Note that when information is stored on disk rather than in memory, the performance of all methods would be affected. In such case, the I/O operations to the storage would dominate the performance of query answering, and it would be difficult to compare the efficiency of these methods. Therefore, in the experiment, we assume that all the information, including geometries of regions, MBRs, constraints, and rectangles are stored in memory. Specifically, the constraints are stored in a hash table indexed by the identifiers of variables, and the rectangles are stored as explained before. The experiment was conducted on a computer with Intel® Core™ i3 1.6 GHz CPU and 16 GB RAM.

Figure 6.7 presents the results of $10{,}000$ random pairs for the following datasets: SEU (which has the largest IMI) and CS+SC (which has the largest number of variables). The queries only involve pairs of variables whose MBRs intersect, and the $10{,}000$ pairs are chosen by randomly sampling in the set of all such pairs. This is motivated by the observation that when geometric information is available, it is easy to apply a pre-test for intersection of the MBRs by all methods. For these two datasets, Label exhibits a clearly better performance than Direct and is reasonably efficient compared to Non-**DC**, which simply needs to retrieve the constraint from a hash table. The median query time for Label is about $3{,}000$ns for SEU and about $2{,}000$ns for CS+SC, while for Direct it is around $500{,}000$ns for both datasets and 500ns for Non-DC. In fact, in both datasets there are about 3000 queries for which the Direct method needs more than 10^6ns. Note that the fact that Label generates more rectangles for the SEU dataset than for the CS+SC dataset translates into a higher query time for the former dataset.

[4]http://www.vividsolutions.com/jts/JTSHome.htm
[5]Using Weight would have similar results and we omitted it here.



(a) (b)

Figure 6.7: (a) Query times on SEU dataset. (b) Query times on CS+SC dataset.

As we discussed in Section 6.2.3, the major procedure of query answering using pseudo-solutions is to search the pseudo-solution arrays and the partial solution arrays. Therefore, the query time for a pair of variables depends on the number of pseudo-solutions (which is the same as the number of clusters) and the number of partial solutions (where the variables appear) for a pseudo-solution. Generally, when these numbers increases, both the storage size and the query time will increase. In the following, we illustrate this on the BA QCNs generated by BA model as in Figure 6.6(a) for the case of **EC**. These QCNs have 500 variables each, and there is only one cluster for each QCN, and thus one pseudo-solution. This is desirable, as we want to fix one factor to see the effect of another factor on query time. Here we only fix the number of pseudo-solution, and the case for fixing the number of partial solutions is similar and will not be considered. We randomly sampled $1{,}000$ queries for each QCN. Figure 6.8 shows the means and medians of the query times. As shown in the figure, the mean and median query times scale about linearly in the average number of rectangles per variable, which is a confirmation of our analysis in Section 6.2.3.

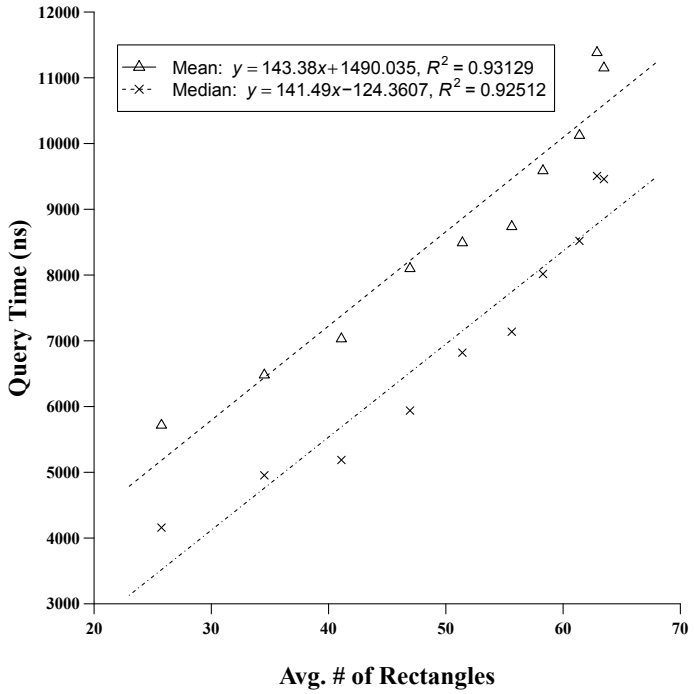


Figure 6.8: Query times on datasets of increasing storage size.

6.5 Summary and Discussion

In the previous sections, we have seen that the pseudo-solution representation can compactly encode the RCC8 information while supporting efficient retrieval of the relations between two given entities. In particular, it works well on large datasets of regions where the number of non-**DC** constraints is relatively large. These cases are the weak points of the previous methods such as MA and its derivatives, and the Non-**DC** method as well. For real-world datasets, under certain optimisations, i.e. Label and Weight, the resulting representation has a size that can be considered as linear in the number of spatial entities. Query answering on the representation has also been shown to be much more efficient than direct computation by using geometric shapes. In fact, the efficiency of query answering is mainly determined by the aver-

age number of rectangles in the representation, while the complexity of the original regions or the number of entities have little effect. As a result, the pseudo-solution representation can scale to very large number of regions (e.g. the dataset CS+SC has more than $100{,}000$ regions), while query answering remains efficient.

In the experiments, we have also seen that there are "degenerate" cases, e.g. where **EC** relation is dominant. In these cases, the pseudo-solution representation might become quadratic. Nevertheless, such cases seem not to be common in real-world applications. Besides, the clustering method could have a negative effect on the performance. This is due to the fact that some variables will be repeatedly clustered into various clusters and the algorithm needs to generate rectangles for them in each cluster. Moreover, when the number of non-**DC** constraints becomes very large, the clustering method might fail. The reason is that in such case there could be a large number of regions that have a common point and will need to be clustered into the same cluster, which in turn makes it impossible to reduce the number of regions in the cluster for the clustering method. Furthermore, the current clustering method cannot deal with pure qualitative input, because it relies on the geometric information. For clustering variables in a QCN, existing algorithms (e.g. [51, 115]) to partition or decompose graphs might be helpful.

The current pseudo-solution method can only handle complete basic RCC8 information. However, the idea of using pseudo-solutions is promising and opens up several areas for future work. For example, this idea can be applied to compactly encoding QCNs for other calculi. Specifically for a complete basic QCN over PA or IA, we can directly make use of a solution of it, which consists of linear number of time points or intervals. From these time points or intervals, the PA or IA relation can be easily calculated. This makes the repre-

sentation of the QCN linear in the number of variants. We might also be able to encode complete basic CDC QCNs. In addition, this idea might be generalised for RCC8 QCNs that contain non-basic relations.

Finally, with the pseudo-solution method and the previous MA method, for various datasets with different properties, we can encode both topological and directional information more compactly and support query answering of relations efficiently. Table 6.13 and Table 6.14 briefly summarised the approaches to representing and retrieving RCC8 or CDC relations discussed in this thesis. By compactly encoding the spatial information, these methods greatly improve the feasibility of building a comprehensive knowledge base of qualitative spatial/temporal information, and will play an important role in applying QSTR to real-world applications.

Table 6.13: Summary of approaches to representing and retrieving RCC8 relations.

Method	Storage Size	Query Time	Useful Case
MA	Less than Complete	Shorter than Complete, Similar to Direct	Usually not as useful as Non-**DC**, but can be used for datasets with a mixture of RCC8 and CDC relations
Spatial-Clustering Index	Larger than MA	Similar to MA	Not more useful than MA
Complete (Database)	Largest	Long, due to database I/O	Small datasets
Non-**DC** (Hash Table)	The least among the first four methods	Dominating the first four methods and MA Derivatives	Large datasets with many **DC** relations
MA Derivatives	Less than MA, sometimes less than Non-**DC**	Shorter than Complete, but may be longer than Direct	Usually not as useful as Non-**DC**
Pseudo-Solution	Less than Non-**DC**	Shorter than Direct, longer than Non-**DC**	Large datasets with many non-**DC** relations
Direct	Storage of regions, can be large	Long when MBRs intersect	When regions are in simple shape

Table 6.14: Summary of approaches to representing and retrieving CDC relations.

Method	Storage Size	Query Time	Useful Case
MA	Less than Complete	Shorter than Direct and Complete	Large dataset where many MBRs do not intersect, or large datasets of mixed RCC8 and CDC relations
Spatial-Clustering Index	Larger than MA	Similar to MA	Not more useful than MA
Complete (Database)	Largest	Long, due to database I/O	Small datasets
MC	Less than MA, sometimes less than Non-DC	Shorter than Complete, but may be longer than MA	Similar to MA
Direct	Storage of regions, can be large	Long when MBRs intersect	When regions are in simple shape

Chapter 7

Conclusion

The capability to handle qualitative spatial and temporal information is desirable in many real-world applications, such as the intelligent personal assistants with growing popularity and geographic information systems. Building a comprehensive knowledge base of qualitative spatial and temporal information is a generalised way to help applications accumulate and process such information. QSTR provides techniques to such a knowledge base for this purpose. QSTR represents the spatial and temporal information between entities as relations, and uses QCNs to gather the relations and to encode connections between relations. The QCN representation adds a qualitative information layer to applications and knowledge bases, and qualitative reasoning techniques help to exploit the QCN for various tasks.

However, there are several important problems that limit the applicability of QSTR to practical circumstances. Specifically, because of the large amount of information to handle, the size of the QCN representation could be so large that the techniques that rely on it become too slow, and the information becomes too costly or even infeasible to be encoded. This calls for efficient reasoning and concise representation. In this thesis, we have provided several solutions

to the corresponding problems.

7.1 Thesis Contributions

Qualitative reasoning techniques on QCNs would be more efficient and more powerful if one can better understand the properties of the relations in the QCNs. The related question we answered in this thesis is what kind of subclasses of qualitative relations are sufficient for algorithms to do qualitative reasoning more efficiently by making use of sparse structures. We proposed the concept of distributive subalgebras to characterise a family of subclasses of relations, and identified maximal ones in various qualitative calculi, i.e. PA, IA RCC5/8, CRA, and RA. Some of the maximal distributive subalgebras in fact coincide with previously identified important subclasses in QSTR, such as the subclasses of convex IA relations [81] and convex RCC8 relations [17]. For distributive subalgebras, we showed that tasks, including the consistency problem, the minimal labelling problem, and the (weakly) globally consistency problem, can be accomplished with efficient algorithms, including DPC (only for the consistency problem), PPC (and PC), and DPC+. Moreover, with the properties of distributive subalgebras, the algorithms DPC and DPC+ reduce the number of propagations compared to PC and PPC to accomplish the same tasks, which means that even in cases the structure is not sparse, these two algorithms will still be more efficient than the algorithms PPC and PC for distributive subalgebras.

For concise representation, we first studied the redundancy problem, i.e. how to identify and to remove redundant constraints in a QCN without changing the solution set of the QCN. The general problem was shown to be co-NP-complete. For tractable subclasses of PA, IA, CRA, RA, and RCC5/8, con-

structing a prime subnetwork can be accomplished in $O(n^4)$ or $O(n^5)$ time, and there can be more than one prime subnetworks. Distributive subalgebras are helpful for solving this problem more efficiently. For any QCN over distributive subalgebras of PA and RCC5/8, we showed that the set of non-redundant constraints (i.e. the core) in the QCN is exactly the unique prime subnetwork of it. On the other hand, we provided examples showing that the results generally do not hold for distributive subalgebras of IA, RA, or CRA, neither for non-distributive subalgebras of RCC5/8 or PA. For a QCN over any distributive subalgebra of PA and RCC5/8, we proposed an $O(n^3)$ algorithm based on the a-closure (i.e. the $\mathbf{PC}$ subnetwork) to construct the core, which was further improved in [113] by using the $\mathbf{PPC}$ subnetwork. With more concise QCNs, the tasks that are sensitive to the number of constraints will become more efficient, such as topological adjustment of geometric data according to the constraints and comparison of QCNs.

In spite of its ability to simplify QCNs, we noticed that the prime subnetwork representation is not suitable for the task of retrieving the relations. This is because it requires qualitative reasoning on the QCN, which is in $O(n^3)$ worst case time (n is the number of variables). Even for a small number of variables the retrieval is too slow. Therefore, we proposed alternative techniques to compactly represent the qualitative information to support efficient retrieval of the relations.

The first technique is the MA method. The idea of MA is that, for RCC8, when the MBRs of two regions do not have a common point, then the RCC8 relation between the regions and that between the MBRs are both **DC**; for CDC, when the MBRs of two regions do not have a common interior point, then the CDC relation between the regions can be calculated either from the MBRs of them or from the MBRs of their connected components. In this way, the only

stored RCC8 (or CDC) relations are the ones between the regions whose MBRs have common point (or common interior point). The number of stored RCC8 or CDC relations by a method is called the *qualified size* of the dataset for the specific method. Compared to the previous approaches proposed in [43], we proved that the qualified size for MA is at most as large as that for those approaches. In real-world datasets, especially for the case of CDC relations, the advantage of MA over these approaches and the complete QCN representation becomes more significant. For retrieval relation of two given regions, we developed the algorithm MAQ based on the compact representation by MA. We demonstrated that for CDC relations MAQ is actually much more efficient than direct computation from geometric representations. When the relations are stored in database, MAQ is also more efficient than the approaches in [43] and the complete QCN representation, because it requires less I/O operations. We also proposed several derivatives of MA that are based on more careful examination of the relations between the MBRs. For some datasets where the average intersection degree is relatively larger, these derivatives can have a higher reduction rate, at the expense of higher computation time to retrieve the relations.

Note that for RCC8, when there is a large number of non-**DC** constraints for the dataset, the MA methods and its derivatives, as well as the Non-**DC** method that only stores non-**DC** constraints, will still need to store a large number of constraints. To represent the RCC8 information more compactly for this kind of datasets, we proposed another technique, i.e. the pseudo-solution representation. For an RCC8 scenario or a set of regions that induce an RCC8 scenario, the idea of this technique is to assign a small number of axis-aligned rectangles to each of the spatial entities, such that the RCC8 relation between any two entities can be calculated from two rectangles for these regions. If on average the

number of rectangles is small for each entity, then the resulting representation will be very compact. Moreover, since the retrieval of relations is based on rectangles, it remains very efficient. We showed on real-world datasets of regions that the pseudo-solution representation can encode the RCC8 relations more compactly than the other approaches, particularly for the datasets where the average number of non-**DC** constraints is very large. We also demonstrated its scalability with respect to the number of non-**DC** constraints, where the pseudo-solution representation shows a sub-linear performance in the number of non-**DC** constraints. For relation retrieval, both on datasets with large average number of non-**DC** constraints and on datasets with large number of regions, the retrieval based on the pseudo-solution representation exhibits much better performance than direct computation from geometric representations, and is reasonably efficient compared to simple table look-up of relations.

All of these results contribute to building a comprehensive knowledge base that helps real-world applications to handle qualitative spatial and temporal information. With more efficient reasoning techniques, such a knowledge base can extract qualitative information from large-scale data and respond to requests by users with less latency. By simplifying the QCN representation, it can serve as a more concise interface of qualitative information. By representing the qualitative information more compactly, it will not only save storage space, but also be scalable to very large datasets without significant decrease in the efficiency to provide qualitative information.

7.2 Future Perspectives

The results in the thesis have shown the potential of applying QSTR to real-world applications to deal with large-scale data. These results lead to several

meaningful future research directions.

7.2.1 Extension of Efficient Algorithms

Distributive subalgebras have many useful properties and algorithms based on these properties have shown promising performance. One natural question is how we can extend these algorithms to deal with larger subclasses or even with the whole qualitative calculus. For example, the traditional backtracking algorithm [101, 112] for checking the consistency of a QCN uses PC or PPC as forward checking of the feasibility of a branch. Note that DPC is more efficient than both of them for checking the consistency of a QCN over a distributive subalgebra. However, it will be inefficient if we directly apply DPC each time when we need to forward check (look ahead) a constraint on the branch, as we do not make use of the consistency of the QCN in the former stage. Therefore, it would be interesting to see how we could properly apply DPC to the backtracking scheme. One promising solution is to develop an incremental version of DPC.

Incrementally applying the algorithms would also help deal with dynamic information, which is very common in real-world applications. For example, in [48, 109, 110], the authors have investigated how to achieve $\mathbf{PC}$ and $\mathbf{PPC}$ with incremental versions of PC and PPC. As we have shown in the thesis, for static QCNs over distributive subalgebras, DPC+ can achieve $\mathbf{PC}$ and $\mathbf{PPC}$ more efficiently than these two algorithms. If we have an incremental version of DPC+, then we can deal with dynamic information in larger scale.

The prime subnetwork technique might also improve the efficiency of qualitative reasoning on dynamic information. By simplifying the structure of the initial QCNs, further update and reasoning on the QCN might be easier. For example, to generate an assignment for the variables so that the values satisfy

the dynamically updated constraints, we might need to check consistency on the changing QCN repeatedly, where a simplification of the QCN will greatly improve the efficiency. How to integrate the prime subnetwork technique with algorithms like DPC and DPC+ would be an interesting challenge.

7.2.2 Extension of Compact Representation

The techniques for compact representation in this thesis only deal with basic relations. In real-world applications, there are also many cases of incomplete or indeterminate information, where the regions might have vague boundaries, or the relations might not be basic (see, e.g., [23, 54, 54]). Being able to deal with this kind of information will greatly increase the scope of the techniques. One possible extension is to introduce "vague" MBRs or axis-aligned rectangles. The "vague" MBRs or rectangles could consist of two rectangles, one of which contains the other. This is like the "egg-yolk" representation in [23]. The incomplete or indeterminate information can then be encoded by the relations between the vague MBRs or rectangles. For example, to encode the constraint $(u\{\mathbf{EC}, \mathbf{PO}\}v)$, we can assign to u the rectangles r_1 and r_2 and to v the rectangle r_3 such that $r_1\mathbf{NTPP}r_2$, $r_1\mathbf{EC}r_3$, and $r_2\mathbf{PO}r_3$.

In addition, it will be an important improvement if we can answer more complex queries with compact representation. On one hand, the current techniques can efficiently retrieve the relation between any pair of variables/regions. On the other hand, it is also vital to deal with the task of retrieving regions that satisfy specific relations with some given regions. One possible approach is to apply indexing techniques such as the variants of R-trees to MBRs and rectangles of compact representation, such that the candidate regions can be quickly filtered out by using the corresponding MBRs or rectangles.

Supporting dynamic information with a compact representation is another

useful extension. For instance, we need to update the representation if some relations are changed or some new variables are added. Note that the changes are usually local, so it is possible that we only need to adjust a small portion of the current representation. How to properly adjust the representation would be worth investigating further.

Finally, repairing the current compact representation techniques for the degenerate cases might lead to substantial improvements. For example, as we have mentioned before, when the number of **EC** constraints becomes large, the number of rectangles generated by the pseudo-solution method could be $O(n^2)$ many. This is due to the shape and dimension of the rectangle. By using a slightly more complex shape (e.g. rectangles with holes) or increasing the dimension, not only the **EC** constraints could be better captured, but also the overall representation might be improved significantly.

Appendix A

Maximal Distributive Subalgebras

In the following, we list the maximal distributive subalgebras of PA, IA, RCC5 and RCC8.

A.1 Maximal Distributive Subalgebras of PA

The closure of basic relations of PA contains 4 non-empty relations

$$\widehat{\mathsf{B}}_{\mathrm{PA}} = \{<, >, =, \star\}. \tag{A.1}$$

One of the maximal distributive subalgebras contains 6 non-empty relations

$$<, >, =, \star, \leq, \geq, \tag{A.2}$$

which is exactly the subclass $\mathcal{C}_{\mathrm{PA}}$ of convex PA relations [122].

The other one contains 5 non-empty relations

$$<, >, =, \star, \neq, \tag{A.3}$$

which is exactly the subclass $\mathcal{S}_{\mathrm{PA}}$ identified in [3].

A.2 Maximal Distributive Subalgebras of IA

The closure of basic IA relations, $\widehat{\mathsf{B}}_{\mathrm{IA}}$, contains 29 non-empty relations (see Table A.1). Our computation shows that IA has two maximal distributive subalgebra, one contains additional 53 non-empty relations, shown in Table A.2, which is exactly the subclass $\mathcal{C}_{\mathrm{IA}}$ of convex IA relations; the other contains

additional 52 non-empty relations, shown in Table A.3, which is exactly the subclass $\mathcal{S}_{\mathrm{IA}}$ identified in [3].

Table A.1: The closure of basic IA relations, $\widehat{\mathrm{B}}_{\mathrm{IA}}$, contains 29 non-empty relations.

{fi}	{f}	{si}	{s}	{mi}	{m}	{oi}	{o}	{di}	{d}	{bi}	{b}	{eq}
{f,fi,eq}			{s,si,eq}			{di,o,fi}			{d,oi,f}			
{d,o,s}			{di,oi,si}			{bi, di, oi, mi, si}			{b, di, o, m, fi}			
{b,o,m}			{bi,oi,mi}			{bi, d, oi, mi, f}			{b, d, o, m, s}			
{bi, d, di, o, oi, mi, s, si, f, fi, eq}						{b, d, di, o, oi, m, s, si, f, fi, eq}						
{d, di, o, oi, s, si, f, fi, eq}						{b, bi, d, di, o, oi, m, mi, s, si, f, fi, eq}						

Table A.2: Additional relations contained in $\mathcal{C}_{\mathrm{IA}}$.

{fi, eq}	{di, fi}	{d, di, o, oi, mi, s, si, f, fi, eq}
{f, eq}	{di, si}	{d, di, o, oi, m, s, si, f, fi, eq}
{si, eq}	{di, si, fi, eq}	{d, di, o, oi, m, mi, s, si, f, fi, eq}
{s, eq}	{di, oi, si, f, fi, eq}	{bi, mi}
{oi, f}	{di, oi, mi, si}	{bi, oi, mi, f}
{oi, si}	{di, oi, mi, si, f, fi, eq}	{bi, oi, mi, si}
{oi, si, f, eq}	{di, o, s, si, fi, eq}	{bi, oi, mi, si, f, eq}
{oi, mi}	{di, o, m, fi}	{bi, di, oi, mi, si, f, fi, eq}
{oi, mi, f}	{di, o, m, s, si, fi, eq}	{bi, d, oi, mi, s, si, f, eq}
{oi, mi, si}	{d, f}	{bi, d, di, o, oi, m, mi, s, si, f, fi, eq}
{oi, mi, si, f, eq}	{d, s}	{b, m}
{o, fi}	{d, s, f, eq}	{b, o, m, fi}
{o, s}	{d, oi, s, si, f, eq}	{b, o, m, s}
{o, s, fi, eq}	{d, oi, mi, f}	{b, o, m, s, fi, eq}
{o, m}	{d, oi, mi, s, si, f, eq}	{b, di, o, m, s, si, fi, eq}
{o, m, fi}	{d, o, s, f, fi, eq}	{b, d, o, m, s, f, fi, eq}
{o, m, s}	{d, o, m, s}	{b, d, di, o, oi, m, mi, s, si, f, fi, eq}
{o, m, s, fi, eq}	{d, o, m, s, f, fi, eq}	

A.3 Maximal Distributive Subalgebras of RCC5

For RCC5, the closure of basic relations $\widehat{\mathrm{B}}_5$ contains 12 non-empty relations. These are the five basic relations, and 7 other relations (see Table A.4).

Table A.3: Additional relations contained in $\mathcal{S}_{\mathrm{IA}}$.

{f, fi}	{bi, d, di, o, oi, s, si}	{b, d, di, o, oi, m, s, si}
{s, si}	{b, bi, d, di, o, oi}	{bi, d, di, o, oi, s, si, f, fi, eq}
{di, oi}	{bi, d, di, o, oi, mi}	{b, bi, d, di, o, oi, f, fi}
{di, o}	{bi, d, di, o, oi, mi, f, fi}	{b, bi, d, di, o, oi, s, si}
{d, oi}	{bi, d, di, o, oi, mi, s, si}	{b, bi, d, di, o, oi, s, si, f, fi, eq}
{d, o}	{bi, d, di, o, oi}	{b, bi, d, di, o, oi, mi}
{b, o}	{d, di, o, oi, f, fi}	{b, bi, d, di, o, oi, mi, f, fi}
{b, di, o}	{bi, d, di, o, oi, f, fi}	{b, bi, d, di, o, oi, mi, s, si}
{b, d, o}	{b, di, o, m}	{b, bi, d, di, o, oi, mi, s, si, f, fi, eq}
{bi, oi}	{d, di, o, oi, s, si}	{b, bi, d, di, o, oi, m}
{bi, di, oi}	{b, d, o, s}	{b, bi, d, di, o, oi, m, f, fi}
{bi, di, oi, si}	{b, d, o, m}	{b, bi, d, di, o, oi, m, s, si}
{bi, di, oi, mi}	{b, d, di, o, oi}	{b, bi, d, di, o, oi, m, s, si, f, fi, eq}
{bi, d, oi}	{b, d, di, o, oi, f, fi}	{b, bi, d, di, o, oi, m, mi}
{bi, d, oi, f}	{b, d, di, o, oi, s, si}	{b, bi, d, di, o, oi, m, mi, f, fi}
{bi, d, oi, mi}	{b, d, di, o, oi, m, f, fi}	{b, bi, d, di, o, oi, m, mi, s, si}
{d, di, o, oi}	{b, d, di, o, oi, m}	{b, d, di, o, oi, s, si, f, fi, eq}
{b, di, o, fi}		

Table A.4: Relations contained in $\widehat{\mathsf{B}}_5$.

$$\{\mathbf{DR}\} \quad \{\mathbf{PO}\} \quad \{\mathbf{PP}\} \quad \{\mathbf{PP}^{-1}\} \quad \{\mathbf{EQ}\}$$
$$\{\mathbf{PO}, \mathbf{PP}\} \quad \{\mathbf{PO}, \mathbf{PP}^{-1}\} \quad \{\mathbf{PO}, \mathbf{PP}, \mathbf{PP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{DR}, \mathbf{PO}, \mathbf{PP}\} \quad \{\mathbf{DR}, \mathbf{PO}, \mathbf{PP}^{-1}\} \quad \{\mathbf{DR}, \mathbf{PO}\} \quad \star.$$

The first maximal distributive subalgebra, denoted by $\mathcal{D}^5_{14}$, contains in addition the following two more relations except the relations in $\widehat{\mathsf{B}}_5$

$$\{\mathbf{PP}, \mathbf{EQ}\}, \{\mathbf{PP}^{-1}, \mathbf{EQ}\}.$$

The second maximal distributive subalgebra, denoted by $\mathcal{D}^5_{20}$, contains in addition eight more relations as shown in Table A.5.

Table A.5: Additional relations contained in $\mathcal{D}^5_{20}$.

$$\{\mathbf{PO}, \mathbf{EQ}\} \quad \{\mathbf{PO}, \mathbf{PP}, \mathbf{EQ}\} \quad \{\mathbf{PO}, \mathbf{PP}, \mathbf{PP}^{-1}\} \quad \{\mathbf{PO}, \mathbf{PP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{DR}, \mathbf{PO}, \mathbf{PP}, \mathbf{PP}^{-1}\} \quad \{\mathbf{DR}, \mathbf{PO}, \mathbf{PP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{DR}, \mathbf{PO}, \mathbf{EQ}\} \quad \{\mathbf{DR}, \mathbf{PO}, \mathbf{PP}, \mathbf{EQ}\}.$$

It is easy to see that both $\mathcal{D}^5_{14}$ and $\mathcal{D}^5_{20}$ are contained in $\mathcal{H}_5$, the maximal tractable subclass of RCC5 identified in [65, 100].

A.4 Maximal Distributive Subalgebras of RCC8

For RCC8, the closure of basic relations $\widehat{\mathsf{B}}_8$ contains 37 non-empty relations as shown in Table A.6.

Table A.6: Relations contained in $\widehat{\mathsf{B}}_8$.

$$\{\mathbf{PO}, \mathbf{TPP}\} \quad \{\mathbf{PO}, \mathbf{TPP}^{-1}\} \quad \{\mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}\}$$
$$\{\mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\} \quad \{\mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{EQ}\} \; \{\mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\} \; \{\mathbf{TPP}, \mathbf{NTPP}\} \quad \{\mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\}$$
$$\{\mathbf{EC}, \mathbf{PO}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}^{-1}\} \; \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}\}$$
$$\{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{EQ}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{DC}, \mathbf{EC}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}^{-1}\}$$
$$\{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\}$$
$$\{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{EQ}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\} \quad \star.$$

The first maximal distributive subalgebra, denoted by $\mathcal{D}^8_{41}$, contains in addition the following four relations

$$\{\mathbf{TPP}, \mathbf{EQ}\} \quad \{\mathbf{TPP}, \mathbf{NTPP}, \mathbf{EQ}\} \quad \{\mathbf{TPP}^{-1}, \mathbf{EQ}\} \quad \{\mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}.$$

This distributive subalgebra turns out to be exactly the class of convex RCC8 relations identified in [17]. The second maximal distributive subalgebra, denoted by $\mathcal{D}_{64}^{8}$, contains in addition 27 more relations shown in Table A.7. It

Table A.7: Additional relations contained in $\mathcal{D}_{64}^{8}$.

$$\{\mathbf{PO}, \mathbf{EQ}\} \quad \{\mathbf{PO}, \mathbf{TPP}, \mathbf{EQ}\} \quad \{\mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{EQ}\} \quad \{\mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}\}$$
$$\{\mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{EQ}\} \quad \{\mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\} \quad \{\mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}\}$$
$$\{\mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{EQ}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{EQ}\}$$
$$\{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{EQ}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{EQ}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}\}$$
$$\{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\} \quad \{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}\}$$
$$\{\mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\}$$
$$\{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{EQ}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{EQ}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}$$
$$\{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{EQ}\} \quad \{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}\}$$
$$\{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\}$$
$$\{\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\}.$$

is easy to check that both $\mathcal{D}_{41}^{8}$ and $\mathcal{D}_{64}^{8}$ are contained in $\mathcal{H}_8$, one of the three maximal subclasses of RCC8 identified in [98].

Appendix B

Supplementary Proofs

B.1 Proofs for Chapter 3

Theorem 3.10. *RA has exactly four maximal distributive subalgebras, which are the Cartesian products of the two maximal distributive subalgebras of IA.*

The following lemma is useful in the proof of the above theorem to transform the intersections, weak compositions, and converses of RA relations.

Lemma B.1 ([6])**.** *For relations R_1, R_2, R_3, R_4 in RA, we have*

- $(R_1 \otimes R_2) \cap (R_3 \otimes R_4) = (R_1 \cap R_3) \otimes (R_2 \cap R_4)$*;*
- $(R_1 \otimes R_2) \diamond (R_3 \otimes R_4) = (R_1 \diamond R_3) \otimes (R_2 \diamond R_4)$*;*
- $(R_1 \otimes R_2)^{-1} = R_1^{-1} \otimes R_2^{-1}$.

In the following we give the proof of the above theorem.

Proof. For convenience, we write $\mathcal{R}_1$ and $\mathcal{R}_2$ for the two maximal distributive subalgebras $\mathcal{C}_{\mathrm{IA}}$ and $\mathcal{S}_{\mathrm{IA}}$. It is straightforward to show that their Cartesian products $\mathcal{R}_i \otimes \mathcal{R}_j$ $(1 \leq i, j \leq 2)$ are all distributive subalgebras of RA.

In order to show the maximality of $\mathcal{R}_i \otimes \mathcal{R}_j$, suppose R is a relation in RA and $R \notin \mathcal{R}_i \otimes \mathcal{R}_j$. We show that the subalgebra $\{R\} \cup \widehat{(\mathcal{R}_i \otimes \mathcal{R}_j)}$ is not distributive. Let $R_x = \{\alpha \in \mathcal{B}_{\mathrm{IA}} : \exists \beta \in \mathcal{B}_{\mathrm{IA}} \text{ s.t. } (\alpha, \beta) \in R\}$ and define R_y similarly. Note that $R \subseteq R_x \otimes R_y$. There are two cases.

Case 1. $R \subset R_x \otimes R_y$. Then there exist relations $\alpha_0, \beta_0 \in \widehat{\mathsf{B}}_{\mathrm{RA}}$ s.t. $\alpha_0 \in R_x$, $\beta_0 \in R_y$, and $\alpha_0 \otimes \beta_0 \notin R$. Let $S = \alpha_0 \otimes \star$ and $T = \star \otimes \beta_0$. Note that $\widehat{\mathsf{B}}_{\mathrm{RA}} \subset \mathcal{R}_i \otimes \mathcal{R}_j$ and $S, T \in \widehat{\mathsf{B}}_{\mathrm{RA}}$. Then $S, T \in \{R\} \cup \widehat{(\mathcal{R}_i \otimes \mathcal{R}_j)}$. It is easy to see

that $R \cap S \neq \varnothing$, $R \cap T \neq \varnothing$, and $S \cap T \neq \varnothing$, but $R \cap S \cap T = \varnothing$. By Theorem 3.8, this implies that $\{R\} \cup \widehat{(\mathcal{R}_i \otimes \mathcal{R}_j)}$ is not distributive.

Case 2. $R = R_x \otimes R_y$. Then we have either $R_x \notin \mathcal{R}_i$ or $R_y \notin \mathcal{R}_j$. Take $R_x \notin \mathcal{R}_i$ as an example. By the maximality of $\mathcal{R}_i$ in IA, we know that $\widehat{\{R_x\} \cup \mathcal{R}_i}$ is not distributive. By Theorem 3.8, this implies that there exist $R_0, S_0, T_0 \in \widehat{\{R_x\} \cup \mathcal{R}_i}$ that do not satisfy the Helly Property (3.5). Note that for any relation A in IA we have $A \diamond \star = \star$, and hence $R_x \otimes \star \in \widehat{\{R_x \otimes R_y\} \cup (\mathcal{R}_i \otimes \mathcal{R}_j)} = \{R\} \cup \widehat{(\mathcal{R}_i \otimes \mathcal{R}_j)}$. Also, for any $B \in \mathcal{R}_i$, we know $B \otimes \star \in \mathcal{R}_i \otimes \mathcal{R}_j$. Together, for any relation $R_1 \in \widehat{\{R_x\} \cup D_i}$ we have $R_1 \otimes \star \in \{R\} \cup \widehat{(\mathcal{R}_i \otimes \mathcal{R}_j)}$. Therefore, $R_0 \otimes \star$, $S_0 \otimes \star$, and $T_0 \otimes \star$ are all in $\{R\} \cup \widehat{(\mathcal{R}_i \otimes \mathcal{R}_j)}$. However, the three relations $R_0 \otimes \star$, $S_0 \otimes \star$, and $T_0 \otimes \star$ do not satisfy (3.5), which means that $\{R\} \cup \widehat{(\mathcal{R}_i \otimes \mathcal{R}_j)}$ is not distributive.

The above shows the maximality of $\mathcal{R}_i \otimes \mathcal{R}_j$. To show the uniqueness, suppose that $\mathcal{S}$ is a distributive subalgebra. We will show that $\mathcal{S}$ is a subset of $\mathcal{R}_i \otimes \mathcal{R}_j$ for some i, j ($1 \leq i, j \leq 2$).

First, we show that for every $R \in \mathcal{S}$ we have $R = R_x \otimes R_y$. Suppose otherwise. Then there exist $\alpha, \beta \in \widehat{\mathsf{B}}_{\mathrm{IA}}$ s.t. $\alpha \in R_x$, $\beta \in R_y$, and $\alpha \otimes \beta \notin R$. Similar to the proof of the maximality, we can show that both $\alpha \otimes \star$ and $\star \otimes \beta$ are in $\widehat{\mathsf{B}}_{\mathrm{RA}}$ and, hence, in $\mathcal{S}$. The three relations $R, \alpha \otimes \star, \star \otimes \beta$, however, do not satisfy (3.5).

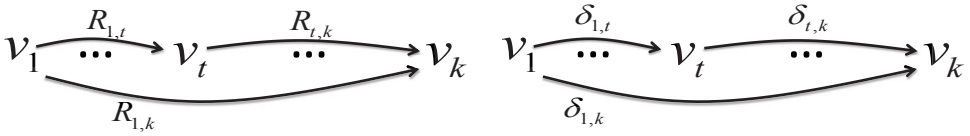
Next, we show that $\mathcal{S}$ is a subset of $\mathcal{R}_i \otimes \mathcal{R}_j$ for some i, j. Write $\mathcal{S}_x = \{R_x : R \in \mathcal{S}\}$ and $\mathcal{S}_y = \{R_y : R \in \mathcal{S}\}$. We assert that $\mathcal{S}_x$ and $\mathcal{S}_y$ are both distributive subalgebras of IA. In fact, we first note that if $R = R_x \otimes R_y \in \mathcal{S}$, then both $R_x \otimes \star$ and $\star \otimes R_y$ are in $\mathcal{S}$. This is because, for instance, $\{\mathsf{eq}\} \otimes \star$ is a relation in $\widehat{\mathsf{B}}_{\mathrm{RA}} \subseteq \mathcal{S}$ and $(R_x \otimes R_y) \diamond (\{\mathsf{eq}\} \otimes \star) = R_x \otimes \star$. Furthermore, it is easy to check that $\{R_x \otimes \star : R_x \otimes R_y \in \mathcal{S}\}$ is a subalgebra and is contained in $\mathcal{S}$, and hence it is distributive. Thus $\mathcal{S}_x$ is a distributive subalgebra of IA and, hence, contained in either $\mathcal{R}_1$ or $\mathcal{R}_2$. The same conclusion applies to $\mathcal{S}_y$. Therefore, $\mathcal{S}$ is a subset of $\mathcal{R}_i \otimes \mathcal{R}_j$ for some i, j. □

Theorem 3.11. *Let $\mathcal{M}$ be a qualitative calculus that satisfies* (3.2) *and* (3.3). *Suppose $\mathcal{S}$ is a distributive subalgebra of $\mathcal{M}$. Then every path consistent QCN over $\mathcal{S}$ is weakly globally consistent and minimal.*

Proof. We first note that, since $\mathcal{M}$ satisfies (3.2), any three relations in $\mathcal{M}$ satisfy the Peircean Law (3.1), and any distributive subalgebra $\mathcal{S}$ of $\mathcal{M}$ is Helly and hence satisfies (3.4).

Suppose $\mathcal{N} = \{v_i R_{ij} v_j : 1 \leq i, j \leq n\}$ is a path consistent QCN over $\mathcal{S}$. Write $V_k = \{v_1, v_2, \ldots, v_k\}$ and $W_t^{k+1} = V_t \cup \{v_{k+1}\}$ for $1 \leq k < n$ and $1 \leq t \leq k$. Let

$\Delta_{V_k} = \{v_i\delta_{ij}v_j : v_i, v_j \in V_k\}$ be a consistent scenario of the restriction of $\mathcal{N}$ on V_k, i.e. $\mathcal{N}|_{V_k} = \{v_iR_{ij}v_j : v_i, v_j \in V_k\}$ (see Figure B.1).



(a) $\mathcal{N}|_{V_k}$ (b) Δ_{V_k}

Figure B.1: Illustration of $\mathcal{N}|_{V_k}$ and Δ_{V_k} in the proof.

We show Δ_{V_k} can be extended to a consistent scenario $\Delta_{V_{k+1}}$ of $\mathcal{N}|_{V_{k+1}}$. Note that any path consistent basic QCN over $\mathcal{M}$ is consistent by the assumption in (3.3).

Let $\widehat{R}_{k+1,i} = \bigcap_{j=1}^{k}(R_{k+1,j} \diamond \delta_{j,i})$ for $i = 1, \ldots, k$. It is easy to see $\widehat{R}_{k+1,i} \subseteq R_{k+1,i}$. Our idea is as follows:

Step 1. Choose an arbitrary basic relation $\delta_{k+1,1}$ in $\widehat{R}_{k+1,1}$.

Step 2. Extend a consistent scenario $\Delta_{W_t^{k+1}}$ to a consistent scenario $\Delta_{W_{t+1}^{k+1}}$ by choosing a certain basic relation $\delta_{k+1,t+1}$ in $\widehat{R}_{k+1,t+1}$, together with the constraints $\{v_i\delta_{i,t+1}v_{t+1} | 1 \leq i \leq t\}$ in Δ_{V_k}. See Figure B.2 for the illustration of $\Delta_{W_t^{k+1}}$ and $\Delta_{W_{t+1}^{k+1}}$.

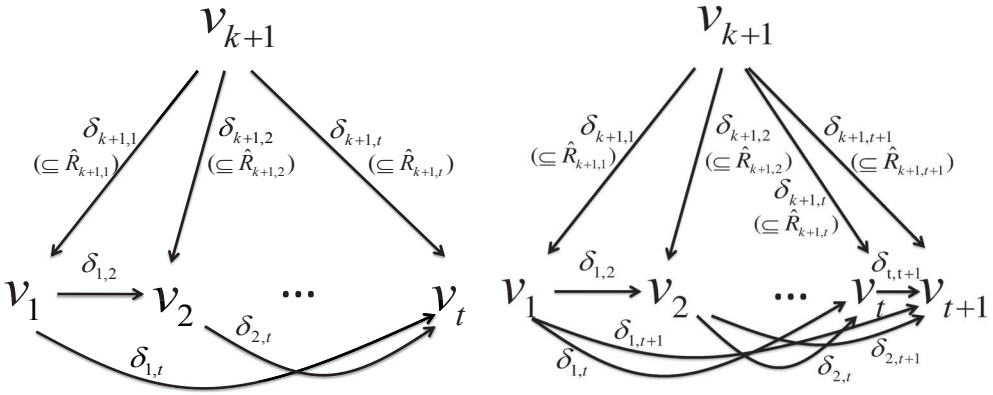
Step 3. Repeat Step 2 for $1 \leq t \leq k-1$ until a consistent scenario $\Delta_{V_{k+1}}$ of $\mathcal{N}|_{V_{k+1}} = \mathcal{N}|_{W_k^{k+1}}$ is obtained.

To show Step 1 can be achieved, we prove that $\widehat{R}_{k+1,i} = \bigcap_{j=1}^{k}(R_{k+1,j} \diamond \delta_{j,i}) \neq \varnothing$ for all $1 \leq i \leq k$. By applying the Peircean Law in (3.1), for $1 \leq j, j' \leq k$ we have

$$\begin{aligned}(R_{k+1,j} \diamond \delta_{ji}) \cap (R_{k+1,j'} \diamond \delta_{j'i}) \neq \varnothing \quad &\Leftrightarrow \quad ((R_{k+1,j'} \diamond \delta_{j'i}) \diamond \delta_{ij}) \cap R_{k+1,j} \neq \varnothing \\ &\Leftrightarrow \quad (R_{j',k+1} \diamond R_{k+1,j}) \cap (\delta_{j'i} \diamond \delta_{ij}) \neq \varnothing.\end{aligned}$$

Since $\mathcal{N}$ is path consistent and the partial scenario Δ_{V_k} is also path consistent, we have $\delta_{j'j} \subseteq R_{j',j} \subseteq R_{j',k+1} \diamond R_{k+1,j}$ and $\delta_{j'j} \subseteq \delta_{j'i} \diamond \delta_{ij}$. Therefore $(R_{j',k+1} \diamond R_{k+1,j}) \cap (\delta_{j'i} \diamond \delta_{ij}) \neq \varnothing$ and hence $(R_{k+1,j} \diamond \delta_{ji}) \cap (R_{k+1,j'} \diamond \delta_{j'i}) \neq \varnothing$ for any $1 \leq j, j' \leq k$. Note that $\mathcal{S}$ is Helly by Proposition 3.8, we know $\widehat{R}_{k+1,i} = \bigcap_{j=1}^{k}(R_{k+1,j} \diamond \delta_{j,i}) \neq \varnothing$ for all $1 \leq i \leq k$.

To show Step 2 can be achieved, we only need to find a basic relation $\delta_{k+1,t+1}$ in $\widehat{R}_{k+1,t+1}$ such that $\Delta_{W_t^{k+1}} \cup \{v_{k+1}\delta_{k+1,t+1}v_{t+1}\}$ is path consistent, for $t = 1, \ldots, k-1$.



(a) $\Delta_{W_t^{k+1}}$ (b) $\Delta_{W_{t+1}^{k+1}}$

Figure B.2: Illustration of $\Delta_{W_t^{k+1}}$ and $\Delta_{W_{t+1}^{k+1}}$ in the proof.

With the following statements, we can show the existence of such $\delta_{k+1,t+1}$.

Statement 1. $\delta_{k+1,i} \diamond \delta_{i,t+1} \cap \widehat{R}_{k+1,t+1} \neq \varnothing$ for any $1 \leq i \leq t$.

Statement 2. $\delta_{k+1,i} \diamond \delta_{i,t+1} \cap \delta_{k+1,j} \diamond \delta_{j,t+1} \neq \varnothing$ for any $1 \leq i, j \leq t$.

In fact, from the above statements and that $\mathcal{S}$ is Helly, we know $(\bigcap_{i=1}^{t}(\delta_{k+1,i} \diamond \delta_{i,t+1})) \cap \widehat{R}_{k+1,t+1} \neq \varnothing$. Thus, there exists a $\delta_{k+1,t+1}$ in $\widehat{R}_{k+1,t+1}$ such that

$$(\bigcap_{i=1}^{t}(\delta_{k+1,i} \diamond \delta_{i,t+1})) \cap \delta_{k+1,t+1} \neq \varnothing. \tag{B.1}$$

To show this $\delta_{k+1,t+1}$ actually extends $\Delta_{W_t^{k+1}}$, we also need to prove that $\Delta_{W_t^{k+1}} \cup \{v_i \delta_{i,t+1} v_{t+1} : 1 \leq i \leq t\} \cup \{v_{k+1} \delta_{k+1,t+1} v_{t+1}\}$ is path consistent. Note that we only need to show $\delta_{k+1,t+1} \subseteq \delta_{k+1,i} \diamond \delta_{i,t+1}$ for any $1 \leq i \leq t$, because $\Delta_{W_t^{k+1}}$ and $\Delta_{V_{t+1}} (\subseteq \Delta_{V_k})$ are both path consistent. This will be true if (B.1) is true. Therefore, in the following, we show that the two statements above are actually true.

For Statement 1, note

$$\begin{aligned}(\bigcap_{j=1}^{k}(R_{k+1,j} \diamond \delta_{j,t+1})) \diamond \delta_{t+1,i} &= \bigcap_{j=1}^{k}(R_{k+1,j} \diamond \delta_{j,t+1} \diamond \delta_{t+1,i}) \\ &\supseteq \bigcap_{j=1}^{k}(R_{k+1,j} \diamond \delta_{ji}).\end{aligned}$$

Then $\widehat{R}_{k+1,t+1} \diamond \delta_{t+1,i} \supseteq \widehat{R}_{k+1,i} \supseteq \delta_{k+1,i} \neq \varnothing$, that is $\delta_{k+1,i} \cap (\widehat{R}_{k+1,t+1} \diamond \delta_{t+1,i}) \neq \varnothing$. By Peircean Law, we have $(\delta_{k+1,i} \diamond \delta_{i,t+1}) \cap \widehat{R}_{k+1,t+1} \neq \varnothing$ for any $1 \leq i \leq t$.

For Statement 2, for any $1 \leq i, j \leq t$ we have

$$\begin{aligned}(\delta_{k+1,i} \diamond \delta_{i,t+1}) \cap (\delta_{k+1,j} \diamond \delta_{j,t+1}) \neq \varnothing \quad &\Leftrightarrow \quad (\delta_{i,k+1} \diamond \delta_{k+1,j} \diamond \delta_{i,t+1}) \cap \delta_{j,t+1} \neq \varnothing \\ &\Leftrightarrow \quad (\delta_{i,k+1} \diamond \delta_{k+1,j}) \cap (\delta_{i,t+1} \diamond \delta_{t+1,j}) \neq \varnothing.\end{aligned}$$

Because $\Delta_{W_t^{k+1}}$ is a (path) consistent scenario, we have $\delta_{ij} \subseteq \delta_{i,k+1} \diamond \delta_{k+1,j}$ for $1 \leq i, j \leq t$. Note $\Delta_{V_{t+1}} (\subseteq \Delta_{V_k})$ is also a (path) consistent scenario, we have $\delta_{ij} \subseteq \delta_{i,t+1} \diamond \delta_{t+1,j}$. Then $(\delta_{i,k+1} \diamond \delta_{k+1,j}) \cap (\delta_{i,t+1} \diamond \delta_{t+1,j}) \supseteq \delta_{ij} \neq \varnothing$ for $1 \leq i, j \leq t$, and hence $(\delta_{k+1,i} \diamond \delta_{i,t+1}) \cap (\delta_{k+1,j} \diamond \delta_{j,t+1}) \neq \varnothing$. □

Proposition 3.21 ([113]). *Let $\mathcal{M}$ be a qualitative calculus that satisfies* (3.2) *and* (3.3). *Suppose $\mathcal{N} = \{v_i R_{ij} v_j : 1 \leq i, j \leq n\}$ is a QCN over a distributive subalgebra $\mathcal{S}$ of $\mathcal{M}$ and $V = \{v_1, ..., v_n\}$. Assume in addition that $G = (V, E)$ is a chordal graph such that $G_{\mathcal{N}} \subseteq G$. Then achieving* **PPC** *on G decides the consistency of $\mathcal{N}$ and results in the same relations on the edges of G as achieving* **PC**.

Before proving the theorem, let us first take a look at a property of chordal graphs from [13].

Lemma B.2 ([13]). *If $G = (V, E)$ is an incomplete chordal graph, then one can add a missing edge $\{u, w\}$ with $u, w \in V$ such that*

- *the graph $G' = (V, E \cup \{\{u, w\}\})$ is chordal graph; and*
- *the graph induced by $X = \{x | \{u, x\}, \{x, w\} \in E\}$ is complete.*

In the following we give the proof of the theorem.

Proof. The proof is similar to the one given for CRC constraints [13, Theorem 3]. This proof does not exploit the conclusion in Theorem 3.11 and is thus slightly different from the one in [113].

Suppose we have a chordal graph $G = (V, E)$ such that $G_{\mathcal{N}} \subseteq G$ and $\mathcal{N}$ is PPC w.r.t. G. We will add to G the missing edges one by one until the graph is complete. To prove the theorem, we show that the relations of the constraints can be computed from the existing ones so that the updated QCN is **PPC** w.r.t. each intermediate graph, including the complete graph.

In the following we assume that the ordering $(v_1, \ldots, v_n)$ is a perfect elimination ordering of the chordal graph G. Denote $S_i = \{v_{n-i+1}, \ldots, v_n\}$, $G_i = G(S_i) = (S_i, E_i)$ where $E_i = \{\{v_r, v_s\} \in E : v_r, v_s \in S_i\}$, and $F_i = \{v_k \in \mathsf{adj}(v_i) : k > i\}$, where $\mathsf{adj}(v_i) = \{v_j : \{v_j, v_i\} \in E\}$.

We add the missing edges one by one to G in the following manner according to Lemma B.2:

1. choose the largest i such that G_i is complete;
2. add a missing edge $\{v_{n-i}, v_j\}$ to G where $v_j \in S_i$;
3. add to $\mathcal{N}$ the constraint $(v_{n-i} R_{n-i,j} v_j)$ where

$$R_{n-i,j} = \bigcap_{v_k \in F_{n-i}} (R_{n-i,k} \diamond R_{k,j}).$$

After adding one edge, we prove that $\mathcal{N}$ is still **PPC** w.r.t. the resulting graph. Figure B.3 illustrates the notations G_i, F_{n-i}, and v_{n-i}.

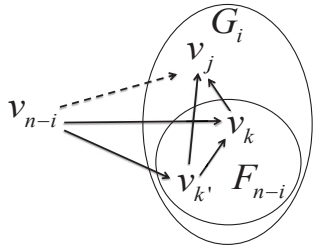


Figure B.3: Illustration for the proof of Proposition 3.21.

First, we show that $R_{n-i,j}$ is non-empty. To show this, by Theorem 3.8, we only need to show that $(R_{n-i,k} \diamond R_{k,j}) \cap (R_{n-i,k'} \diamond R_{k',j}) \neq \varnothing$ for any $v_k \neq v_{k'} \in F_{n-i}$. Such a pairwise intersection is not empty because, by the Peircean Law, we have

$$(R_{n-i,k} \diamond R_{k,j}) \cap (R_{n-i,k'} \diamond R_{k',j}) \neq \varnothing \Leftrightarrow (R_{k,n-i} \diamond R_{n-i,k'}) \cap (R_{k,j} \diamond R_{j,k'}) \neq \varnothing.$$

Since $G(F_{n-i} \cup \{v_{n-i}\})$[1] and G_i are complete and the corresponding subnetworks on them are path consistent, we have $R_{k,k'} \subseteq R_{k,n-i} \diamond R_{n-i,k'}$ and $R_{k,k'} \subseteq R_{k,j} \diamond R_{j,k'}$. This shows that $(R_{k,n-i} \diamond R_{n-i,k'}) \cap (R_{k,j} \diamond R_{j,k'}) \neq \varnothing$ and, hence, $(R_{n-i,k} \diamond R_{k,j}) \cap (R_{n-i,k'} \diamond R_{k',j}) \neq \varnothing$.

We then need to show that the resulting QCN is path consistent. To this end, we only need to consider the three paths $(v_{n-i}, v_j, v_{k'})$, $(v_{n-i}, v_{k'}, v_j)$, and $(v_{k'}, v_{n-i}, v_j)$.

For $(v_{n-i}, v_j, v_{k'})$, note that, for any $k \in F_{n-i}$, we have $R_{n-i,k'} \subseteq R_{n-i,k} \diamond R_{k,k'} \subseteq R_{n-i,k} \diamond R_{k,j} \diamond R_{j,k'}$. Therefore, we have $R_{n-i,k'} \subseteq \bigcap_{k \in F_{n-i}} (R_{n-i,k} \diamond R_{k,j} \diamond R_{j,k'})$. By distributivity, we know $R_{n-i,k'} \subseteq (\bigcap_{k \in F_{n-i}} R_{n-i,k} \diamond R_{k,j}) \diamond R_{j,k'} = R_{n-i,j} \diamond R_{k,j}$.

For $(v_{n-i}, v_{k'}, v_j)$, by the construction of $R_{n-i,j}$, we have $R_{n-i,j} \subseteq R_{n-i,k'} \diamond R_{k',j}$.

[1] Note that $G(F_{n-i} \cup \{v_{n-i}\}) = (F_{n-i} \cup \{v_{n-i}\}, E^*)$, where $E^* = \{\{v_r, v_s\} \in E : v_r, v_s \in F_{n-i} \cup \{v_{n-i}\}\}$.

For $(v_{k'}, v_{n-i}, v_j)$, we need to show that $R_{k',j} \subseteq R_{k',n-i} \diamond R_{n-i,j}$. Note that $R_{n-i,j} = \bigcap_{v_k \in F_{n-i}} (R_{n-i,k} \diamond R_{k,j})$. Because $\mathcal{S}$ is distributive, it is sufficient to show, for each $k \in F_{n-i}$, $R_{k',j} \subseteq R_{k',n-i} \diamond R_{n-i,k} \diamond R_{k,j}$. Because $G(F_{n-i} \cup \{v_{n-i}\})$ is complete and $\mathbf{PC}$, we have $R_{k',k} \subseteq R_{k',n-i} \diamond R_{n-i,k}$. Moreover, because $G(F_{n-i} \cup \{v_j\}) \subseteq G_i$ is complete and $\mathbf{PC}$ by construction and induction, we know $R_{k',j} \subseteq R_{k',k} \diamond R_{k,j} \subseteq R_{k',n-i} \diamond R_{n-i,k} \diamond R_{k,j}$.

Thus, after adding a missing edge to G and add the corresponding constraint to $\mathcal{N}$, $\mathcal{N}$ remains $\mathbf{PPC}$ w.r.t. G. At last we will get the complete graph that is the completion of G. Note that the constraint corresponding to each of the original edges in G is not changed. This finishes the proof. □

Theorem 3.23. *Let $\mathcal{N} = (V, \mathcal{C})$ be a QCN that is defined over a distributive subalgebra of a qualitative calculus that satisfies (3.2) and (3.3), and $\alpha = (v_1, \ldots, v_n)$ an ordering of V. Then,* DPC+ *returns* $(\mathsf{True}, G, \mathcal{N}')$ *if and only if $\mathcal{N}$ is satisfiable, where G is a chordal graph such that $G_{\mathcal{N}} \subseteq G$ and α is a perfect elimination ordering of it, and $\mathcal{N}'$ is the $\mathbf{PPC}$ w.r.t. G subnetwork of $\mathcal{N}$*

Proof. After calling DPC in line 1, $\mathcal{N}$ becomes $\mathbf{DPC}$ w.r.t. α and we get a chordal graph G such that $G_{\mathcal{N}} \subseteq G$ and α is a perfect elimination ordering of it. In what follows, we denote by $\mathcal{N}^{(0)}$ the $\mathbf{DPC}$ network before applying the next steps of DPC+ and by $\mathcal{N}$ the updated network obtained afterwards.

Suppose that $\mathcal{N}_k$ is the restriction of the updated $\mathcal{N}$ on the variables $\{v_{n-k+1}, \ldots, v_{n-1}, v_n\}$, that is, the updated partial QCN after considering the variables $\{v_{n-k+1}, \ldots, v_{n-1}, v_n\}$ in the **for** loop in line 4 of DPC+. Note that $\mathcal{N}_n = \mathcal{N}$. It suffices to show that $\mathcal{N}_k$ is $\mathbf{PPC}$ given that $\mathcal{N}_{k-1}$ is $\mathbf{PPC}$. For simplicity, let $t = n-k+1$. Then to this end, we only need to consider the variables in F_t, and show that $\forall v_i, v_j \in F_t$, we have $R_{it} \neq \varnothing$, $R_{ij} \subseteq R_{it} \diamond R_{tj}$, and $R_{it} \subseteq R_{ij} \diamond R_{jt}$. Figure B.4 illustrates the case.

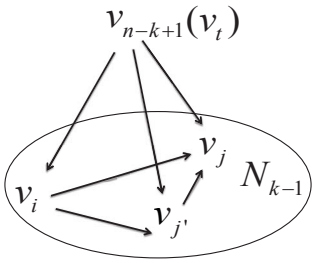


Figure B.4: Illustration for the proof of Theorem 3.23.

To simplify our proof, we first adjust the updating rule in line 7 of DPC+ from $R_{it} \leftarrow \bigcap_{v_j \in F_t} R_{ij} \diamond R_{jt}$ to $R_{it} \leftarrow \bigcap_{v_j \in F_t} R_{ij} \diamond R^{(0)}_{jt}$, where $R^{(0)}_{jt}$ is the relation between v_j and v_t in the original $\mathbf{DPC}$ network, viz., $\mathcal{N}^{(0)}$. We denote the

adjusted algorithm by DPC+*. We will first prove that the conclusion holds for DPC+*.

We first show that $R_{it} \neq \varnothing$ for all $v_i \in F_t$. Note that $\forall v_j, v_{j'} \in F_t$, by **DPC** of $\mathcal{N}^{(0)}$, we have $R_{jj'} \subseteq R^{(0)}_{jj'} \subseteq R^{(0)}_{jt} \diamond R^{(0)}_{tj'}$. By **DPC** of $\mathcal{N}_{t-1}$, we have $R_{jj'} \subseteq R_{ji} \diamond R_{ij'}$. Therefore, $\varnothing \neq R_{jj'} \subseteq R^{(0)}_{jt} \diamond R^{(0)}_{tj'} \cap R_{ji} \diamond R_{ij'}$. By the Peircean Law in 3.1, we have $(R_{ij} \diamond R^{(0)}_{jt}) \cap (R_{ij'} \diamond R^{(0)}_{j't}) \neq \varnothing$. According to the Helly property of distributive subalgebras as in 3.4, we have $R_{it} = \bigcap_{v_j \in F_t} R_{ij} \diamond R^{(0)}_{jt} \neq \varnothing$.

Next, we show that $R_{ij} \subseteq R_{it} \diamond R_{tj}$. Note that as $R_{jt} = \bigcap_{v_{j''} \in F_t} R_{jj''} \diamond R^{(0)}_{j''t}$, we have

$$R_{it} \diamond R_{tj} = (\bigcap_{v_{j'} \in F_t} (R_{ij'} \diamond R^{(0)}_{j't})) \diamond (\bigcap_{v_{j''} \in F_t} (R^{(0)}_{tj''} \diamond R_{j''j})).$$

Because the relations are in a distributive subalgebra, we have

$$R_{it} \diamond R_{tj} = \bigcap_{v_{j'} \in F_t} \bigcap_{v_{j''} \in F_t} (R_{ij'} \diamond R^{(0)}_{j't} \diamond R^{(0)}_{tj''} \diamond R_{j''j}).$$

As $\mathcal{N}^{(0)}$ is **DPC**, we have $R_{j'j''} \subseteq R^{(0)}_{j'j''} \subseteq R^{(0)}_{j't} \diamond R^{(0)}_{tj''}$, and as $\mathcal{N}_{t-1}$ is **PPC**, we have $R_{ij} \subseteq R_{ij'} \diamond R_{j'j''} \diamond R_{j''j}$. Therefore $R_{ij} \subseteq R_{ij'} \diamond R^{(0)}_{j't} \diamond R^{(0)}_{tj''} \diamond R_{j''j}$, $\forall v_{j'}, v_{j''} \in F_t$, and hence $R_{ij} \subseteq R_{it} \diamond R_{tj}$.

Next, we show that $R_{it} \subseteq R_{ij} \diamond R_{jt}$. In fact,

$$\begin{aligned} R_{it} = \bigcap_{v_{j'} \in F_t} (R_{ij'} \diamond R^{(0)}_{j't}) &\subseteq \bigcap_{v_{j'} \in F_t} (R_{ij} \diamond R_{jj'} \diamond R^{(0)}_{j't}) \\ = R_{ij} \diamond (\bigcap_{v_{j'} \in F_t} (R_{jj'} \diamond R^{(0)}_{j't})) &= R_{ij} \diamond R_{jt}. \end{aligned}$$

Since v_j is arbitrary in F_t and $R_{j't} \subseteq R^{(0)}_{j't}$, we also have

$$R_{it} = \bigcap_{v_{j'} \in F_t} (R_{ij'} \diamond R^{(0)}_{j't}) = \bigcap_{v_{j'} \in F_t} (R_{ij'} \diamond R_{j't}). \tag{B.2}$$

Therefore, under the updating rule of DPC+*, we have that $\mathcal{N}_t$ is **PPC** if N_{t-1} is **PPC** and, thus, DPC+* enforces **PPC** on $\mathcal{N}$. Regarding DPC+, its updating rule will update the first R_{it} by using relations in $\mathcal{N}_{t-1}$ and $\mathcal{N}^{(0)}$, and then update the following $R_{i't}$ by using the updated R_{it}, and so on. By induction, we can prove that each updated R_{it} is stronger than $R^{(0)}_{it}$ and weaker than $\bigcap_{v_{j'} \in F_t} R_{ij'} \diamond R^*_{j't}$, where $R^*_{j't}$ is obtained by DPC+*. By (B.2), the relations ob-

tained by DPC+ are the same as those obtained by DPC+*. As DPC+* enforces $\mathbf{PPC}$ on $\mathcal{N}$, we have that DPC+ also enforces $\mathbf{PPC}$ on $\mathcal{N}$. □

B.2 Proofs for Chapter 4

Proposition 4.11. *Let $\mathcal{S}$ be a distributive subalgebra of PA or RCC5/8. Suppose $\mathcal{N}$ is an all-different QCN over $\mathcal{S}$. Assume that (xRy) and (xSy) are the constraints from x to y in $\mathcal{N}$ and $\mathcal{N}_\mathrm{p}$ respectively. Then (xRy) is redundant in $\mathcal{N}$ iff (xSy) is redundant in $\mathcal{N}_\mathrm{p}$.*

Proof. Here we prove the sufficiency part. We focus on the case of RCC5 and RCC8, while following a similar procedure the case for PA can be easily proved.

Write $\mathcal{N}'$ and $\mathcal{N}''$ for $\mathcal{N} \setminus \{(xRy)\}$ and $\mathcal{N}_\mathrm{p} \setminus \{(xSy)\}$ respectively. Suppose (xSy) is redundant in $\mathcal{N}_\mathrm{p}$. Let W be the intersection of the weak compositions of all paths from x to y in $\mathcal{N} \setminus \{(xRy)\}$. To show that (xRy) is redundant in $\mathcal{N}$, by Lemma 4.6, we only need to show $R \supseteq W$.

Recall $S = R \cap W$ by Lemma 3.15. To show $R \subseteq W$, we first show

$$R \cap W \supseteq W \cap \mathbf{O}_l \diamond R \cap R \diamond \mathbf{O}_l, \tag{B.3}$$

where $\mathbf{O}_l$ is either $\mathbf{O}_5$ or $\mathbf{O}_8$ (cf. Lemma 3.14 for definition), according to whether $\mathcal{N}$ is over RCC5 or RCC8.

Because (xSy) is redundant in $\mathcal{N}_\mathrm{p}$, by Corollary 4.9, we know S is the intersection of the weak compositions of all paths from x to y in $\mathcal{N}$ of length ≥ 2.

As we have seen in the proof of Proposition 3.15, there are three types of paths of length ≥ 2. For every path π of Case 1 or 3 (cf. Proposition 3.15), we know $\mathrm{CT}(\pi)$ contains W. Suppose π is a path of Case 2 and $c_i = (xRy)$ for some $1 \leq i \leq s$. Then $\mathrm{CT}(\pi) = \mathrm{CT}(\pi_{<i}) \diamond R \diamond \mathrm{CT}(\pi_{>i})$. Note that if $\pi_{<i}$ ($\pi_{>i}$, respectively) is non-empty, then $\mathrm{CT}(\pi_{<i})$ ($\mathrm{CT}(\pi_{>i})$, respectively) contains $\mathbf{O}_l$ by Lemma 3.14. Either $\pi_{<i}$ or $\pi_{>i}$ is a cycle path. Therefore, $\mathrm{CT}(\pi)$ contains $\mathbf{O}_l \diamond R \cap R \diamond \mathbf{O}_l \cap \mathbf{O}_l \diamond R \diamond \mathbf{O}_l$. In summary, for each path π from x to y in $\mathcal{N}$ with length ≥ 2, we have $\mathrm{CT}(\pi) \supseteq W \cap \mathbf{O}_l \diamond R \cap R \diamond \mathbf{O}_l \cap \mathbf{O}_l \diamond R \diamond \mathbf{O}_l$. Because $\mathbf{O}_l \diamond R \diamond \mathbf{O}_l$ is always the universal relation (as $\mathbf{PO} \diamond R \diamond \mathbf{PO} = \mathbf{PO} \diamond \mathbf{PO} = \star$ (by Lemma 3.14), we know S, as the intersection of the weak compositions of *all* paths from x to y in $\mathcal{N}$ with length ≥ 2, contains $W \cap \mathbf{O}_l \diamond R \cap R \diamond \mathbf{O}_l$. Since $S = R \cap W$, we have (B.3) immediately.

We next show $R \supseteq W$. Because $\mathcal{N}$ is consistent and satisfies (3.8), we know $S = R \cap W$ is neither empty nor $\{\mathbf{EQ}\}$, i.e.,

$$\varnothing \neq R \cap W \neq \{\mathbf{EQ}\}.$$

If **PO** $\in R$, then $\mathbf{O}_l \diamond R \cap R \diamond \mathbf{O}_l \supseteq \mathbf{PO} \diamond \mathbf{PO}$ is the universal relation. That $R \supseteq W$ follows directly from $R \cap W \supseteq W \cap \star = W$.

If **PO** $\notin R$, then **PO** $\notin W$ because **PO** $\in \mathbf{O}_l \diamond R \cap R \diamond \mathbf{O}_l$ and (B.3) holds. We show $R \supseteq W$. We only consider RCC8 relations. The case for RCC5 relations is similar. Suppose R is a relation in a distributive subalgebra of RCC8 such that **PO** $\notin R$ and $R \neq$ **EQ**. By checking the lists of relations in the two maximal distributive subalgebras given in Appendix A.4, R is either a basic relation other than **PO** and **EQ**, or one of the following relations

$$\begin{aligned} &\{\mathbf{TPP}, \mathbf{NTPP}\}, \{\mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\}, \\ &\{\mathbf{DC}, \mathbf{EC}\}, \{\mathbf{TPP}, \mathbf{EQ}\}, \{\mathbf{TPP}^{-1}, \mathbf{EQ}\}, \\ &\{\mathbf{TPP}, \mathbf{NTPP}, \mathbf{EQ}\}, \{\mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}. \end{aligned} \tag{B.4}$$

There are several sub-cases. Suppose R is a basic relation α other than **PO** and **EQ**. We write α^d for the other basic relation such that $\{\alpha, \alpha^d\}$ is a relation in (B.4). For example, $\mathbf{DC}^d = \mathbf{EC}$, $\mathbf{TPP}^d = \mathbf{NTPP}$, and $\mathbf{TPP}^{-1^d} = \mathbf{NTPP}^{-1}$. From the RCC8 composition table we can see

$$\{\alpha, \alpha^d, \mathbf{PO}\} \subseteq \mathbf{PO} \diamond \alpha \cap \alpha \diamond \mathbf{PO} \subseteq \mathbf{O}_8 \diamond \alpha \cap \alpha \diamond \mathbf{O}_8$$

holds for every basic relation α other than **PO** and **EQ**. We assert that $\alpha^d \notin W$ if $R = \{\alpha\}$. This is because, otherwise, we have $\alpha^d \in W \cap \mathbf{O}_8 \diamond R \cap R \diamond \mathbf{O}_8$ and hence by (B.3) $\alpha^d \in R \cap W \subseteq R$. A contradiction. In particular, if α is **DC**, **EC**, **NTPP**, or $\mathbf{NTPP}^{-1}$, then $W = R$. If α is either **TPP** or $\mathbf{TPP}^{-1}$, then we can further show that **EQ** $\in \mathbf{O}_8 \diamond \alpha \cap \alpha \diamond \mathbf{O}_8$ and hence **EQ** $\notin W$. This implies that $W = R$.

Suppose R is $\{\mathbf{DC}, \mathbf{EC}\}$, $\{\mathbf{TPP}, \mathbf{NTPP}, \mathbf{EQ}\}$, or $\{\mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}, \mathbf{EQ}\}$. Note that **PO** $\notin W$, and $\varnothing \neq R \cap W \neq \{\mathbf{EQ}\}$. This shows that W is contained in R.

Suppose R is $\{\mathbf{TPP}, \mathbf{NTPP}\}$ or $\{\mathbf{TPP}^{-1}, \mathbf{NTPP}^{-1}\}$. By (B.3) and **EQ** $\in \mathbf{O}_8 \diamond R \cap R \diamond \mathbf{O}_8$ we know that W does not contain **EQ**. Hence W is contained in R.

Suppose R is $\{\mathbf{TPP}, \mathbf{EQ}\}$. By (B.3) and **NTPP** $\in \mathbf{O}_8 \diamond R \cap R \diamond \mathbf{O}_8$, W cannot contain **NTPP**. This implies that W is contained in R. The case for $R = \{\mathbf{TPP}^{-1}, \mathbf{EQ}\}$ is similar.

In summary, we have $R \supseteq W$ in all cases. In other words, R can be obtained as the intersection of all paths from x to y in $\mathcal{N} \setminus \{(xRy)\}$. Hence (xRy) is redundant in $\mathcal{N}$ by Lemma 4.6. □

B.3 Proofs for Chapter 5

Proposition 5.7. *Given a set of possibly disconnected regions $\mathcal{D} = \{o_1, ..., o_n\}$, then, for RCC8, the qualified size of $\mathcal{D}$ for* MA *is not larger than the qualified sizes of $\mathcal{D}$ for either the grid clustering index or the R-tree clustering index.*

Proof. Suppose the RCC8 relation for two regions $o_i, o_j \in \mathcal{D}$ is calculated and stored by MA. Then their MBRs must intersect, i.e. $\mathsf{mbr}(o_i) \cap \mathsf{mbr}(o_j) \neq \varnothing$. Note that for the grid and the R-tree clustering indexes, the MBRs of regions are used to build the spatial clustering structure, and by the definition of the spatial clustering structure, the tiles in the index completely cover the objects (and, hence, their MBRs), i.e. $\bigcup_{k=1}^{n} \mathsf{mbr}(o_k) \subseteq \bigcup_{l \in J} t_l$.

For the grid clustering index, by the above assumption, we know there must be one index tile t_0 s.t. $t_0 \cap \mathsf{mbr}(o_i) \cap \mathsf{mbr}(o_j) \neq \varnothing$. By the strategy of building a clustering structure of the grid clustering index for RCC8 (see Table 5.1), we know that o_i and o_j are associated with the same clustering structure entry (t_0, C_0). Therefore, according to Step 2(a) for building a spatial clustering index on Page 122. the RCC8 relation between o_i and o_j will be computed and stored by the grid clustering index.

For the R-tree clustering index, we only need to prove that there exist two leaf index tiles t_1 and t_2 such that $t_1 \cap t_2 \neq \varnothing$ and o_i and o_j are associated with t_1 and t_2 respectively.

As shown in Table 5.1, there are two strategies for the R-tree clustering index to associate a region to a leaf index tile. In the first strategy, by the assumption that $\bigcup_{k=1}^{n} \mathsf{mbr}(o_k) \subseteq \bigcup_{l \in J} t_l$, we know that $\mathsf{mbr}(o_i)$ and $\mathsf{mbr}(o_j)$, as well as $\mathsf{mbr}(o_i) \cap \mathsf{mbr}(o_j)$, are covered by several leaf index tiles. Then among the index tiles that cover $\mathsf{mbr}(o_i)$ and $\mathsf{mbr}(o_j)$, there are two index tiles t_1 and t_2 (t_1 might be equal to t_2) such that (i) t_1 and $\mathsf{mbr}(o_i)$ have a common interior point, (ii) t_2 and $\mathsf{mbr}(o_j)$ have a common interior point, and (iii) $t_1 \cap t_2 \cap \mathsf{mbr}(o_i) \cap \mathsf{mbr}(o_j) \neq \varnothing$. This means that o_1 and o_2 are associated with t_1 and t_2, respectively, and t_1 and t_2 are not **DC**, which is the only clustering relation of RCC8. For the R-tree clustering index that is built by the second strategy, again by the assumption that $\bigcup_{k=1}^{n} \mathsf{mbr}(o_k) \subseteq \bigcup_{l \in J} t_l$, we know that there must be two leaf index tiles t_1 and t_2 (t_1 might be equal to t_2) such that o_1 and o_2 are associated with t_1 and t_2, respectively, and $\mathsf{mbr}(o_i) \subseteq t_1$ and $\mathsf{mbr}(o_j) \subseteq t_2$. Therefore $t_1 \cap t_2 \supseteq \mathsf{mbr}(o_i) \cap \mathsf{mbr}(o_j) \neq \varnothing$, i.e. t_1 and t_2 are not in clustering relation **DC**.

From the above discussion, we know the RCC8 relation between o_i and o_j will also be calculated and stored by the R-tree clustering index. □

Proposition 5.8. *Given a set of possibly disconnected regions $\mathcal{D} = \{o_1, ..., o_n\}$, for CDC the qualified size of $\mathcal{D}$ for* MA *is not larger than the qualified sizes for*

either the grid or the R-tree clustering indexes.

Proof. Suppose the CDC relation from region $o_i \in \mathcal{D}$ to region $o_j \in \mathcal{D}$ is calculated and stored by MA. This implies that their MBRs have a common interior point, i.e. $(\mathsf{mbr}(o_i))^\circ \cap (\mathsf{mbr}(o_j))^\circ \neq \varnothing$. Note that we have $\bigcup_{k=1}^{n} \mathsf{mbr}(o_k) \subseteq \bigcup_{l \in J} t_l$.

For the grid clustering index, by the above assumption, we know there must be one index tile t_0 s.t. $t_0 \cap (\mathsf{mbr}(o_i))^\circ \cap (\mathsf{mbr}(o_j))^\circ \neq \varnothing$. By the grid clustering indexing strategy of building a clustering structure for CDC we know o_i and o_j are both associated with t_0. Therefore, the CDC relations between o_i and o_j will be calculated and stored by the grid clustering index.

For the R-tree clustering index, we only need to prove that there exist two leaf index tiles t_1, t_2 such that $t_1 \cap t_2 \neq \varnothing$ and o_i and o_j are associated with t_1 and t_2 respectively.

We consider the two strategies of building the R-tree index for leaf index tiles (Table 5.1). For the first strategy, like the above discussion for the grid clustering index, we know there is a leaf index tile t_0 such that o_i and o_j are both associated with t_0. For the second strategy, we know that there exist two leaf index tiles t_1 and t_2 (t_1 might be equal to t_2) such that o_1 and o_2 are associated with t_1 and t_2, respectively, and $\mathsf{mbr}(o_i) \subseteq t_1$ and $\mathsf{mbr}(o_j) \subseteq t_2$. Therefore $t_1^\circ \cap t_2^\circ \supseteq (\mathsf{mbr}(o_i))^\circ \cap (\mathsf{mbr}(o_j))^\circ \neq \varnothing$, i.e. t_1 and t_2 have a common interior point. Thus t_1 and t_2 are not in any CDC clustering relation.

From the above discussion, the CDC relation from o_i to o_j will be calculated and stored by the R-tree clustering index. □

Appendix C

List of Publications

The thesis is related to the following joint work with co-authors.

- Chapter 3 is mainly based on the follow publications.
  - Zhiguo Long and Sanjiang Li: On Distributive Subalgebras of Qualitative Spatial and Temporal. Conference on Spatial Information Theory (COSIT 2015), pp. 354–374.
  - Zhiguo Long, Michael Sioutis, and Sanjiang Li: Efficient Path Consistency Algorithm for Large Qualitative Constraint Networks. 25th International Joint Conference on Artificial Intelligence (IJCAI 2016), pp. 1202–1208.
- The follow publication is also related to Chapter 3.
  - Michael Sioutis, Zhiguo Long, and Sanjiang Li: Efficiently Reasoning about Qualitative Constraints through Variable Elimination. 9th Hellenic Conference on Artificial Intelligence (SETN 2016), pp. 1–10. (Best Paper Award)
- Chapter 4 is mainly based on the following publication, which also contains results related to Chapter 3.
  - Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both: On Redundant Topological Constraints. Artificial Intelligence, 2015, vol. 225, pp. 51–76.
- The following publication is an extended abstract of the above publication.

  - Matt Duckham, Sanjiang Li, Weiming Liu, and Zhiguo Long: On Redundant Topological Constraints (Extended Abstract). 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014), pp. 618–621.

- Chapter 5 is based on the following publication.

  - Zhiguo Long, Matt Duckham, Sanjiang Li, and Steven Schockaert: Indexing Large Geographic Datasets with Compact Qualitative Representation. International Journal of Geographical Information Science, 2016, vol. 30, no. 6, pp. 1072–1094.

- Chapter 6 is based on the following publication.

  - Zhiguo Long, Steven Schockaert, and Sanjiang Li: Encoding Large RCC8 Scenarios Using Rectangular Pseudo-Solutions. 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016), pp. 463–472.

Below we list the other publications during my PhD candidature that are out of the scope of this thesis.

- Zhiguo Long and Sanjiang Li: A Complete Classification of Spatial Relations Using the Voronoi-Based Nine-Intersection Model, International Journal of Geographical Information Science, 2013, vol. 27, no. 10, pp. 2006–2025.

- Jae Hee Lee, Sanjiang Li, Zhiguo Long, and Michael Sioutis: On Redundancy in Simple Temporal Networks. 22nd European Conference on Artificial Intelligence (ECAI 2016), pp. 828–836. All authors contributed equally.

- Shufeng Kong, Sanjiang Li, Yongming Li, and Zhiguo Long: On Tree-Preserving Constraints. 21st International Conference on Principles and Practice of Constraint Programming (CP 2015), pp. 244–261.

Bibliography

[1] Al-Salman, R. [2014], Qualitative Spatial Query Processing: Towards Cognitive Geographic Information Systems, PhD thesis, University of Bremen, Germany.

[2] Allen, J. F. [1983], 'Maintaining knowledge about temporal intervals', *Communications of the ACM* **26**(11), 832–843.

[3] Amaneddine, N. and Condotta, J.-F. [2012], From path-consistency to global consistency in temporal qualitative constraint networks, *in* 'International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA 2012)', pp. 152–161.

[4] Amaneddine, N. and Condotta, J.-F. [2013], On the minimal labeling problem of temporal and spatial qualitative constraints, *in* 'The Twenty-Sixth International FLAIRS Conference (FLAIRS 2013)', pp. 16–21.

[5] Amaneddine, N., Condotta, J.-F. and Sioutis, M. [2013], Efficient approach to solve the minimal labeling problem of temporal and spatial qualitative constraints, *in* 'International Joint Conference on Artificial Intelligence (IJCAI 2013)', pp. 696–702.

[6] Balbiani, P., Condotta, J.-F. and Fariñas del Cerro, L. [1999], A new tractable subclass of the rectangle algebra, *in* 'International Joint Conference on Artificial Intelligence (IJCAI 1999)', pp. 442–447.

[7] Balbiani, P., Condotta, J.-F. and Fariñas del Cerro, L. [2002], 'Tractability results in the block algebra', *Journal of Logic and Computation* **12**(5), 885–909.

[8] Barabási, A.-L. and Albert, R. [1999], 'Emergence of scaling in random networks', *Science* **286**(5439), 509–512.

[9] Beckmann, N., Kriegel, H., Schneider, R. and Seeger, B. [1990], The R*-tree: An efficient and robust access method for points and rectan-

gles, *in* 'International Conference on Management of Data (ACM SIGMOD 1990)', pp. 322–331.

[10] Bentley, J. L. and Wood, D. [1980], 'An optimal worst case algorithm for reporting intersections of rectangles', *IEEE Transactions on Computers* **C-29**(7), 571–577.

[11] Bessière, C., Isli, A. and Ligozat, G. [1996], Global consistency in Interval Algebra networks: Tractable subclasses, *in* 'European Conference on Artificial Intelligence (ECAI 1996)', pp. 3–7.

[12] Blair, J. R. S. and Peyton, B. [1993], An introduction to chordal graphs and clique trees, *in* 'Graph Theory and Sparse Matrix Computation', Springer, Heidelberg, Germany, pp. 1–29.

[13] Bliek, C. and Sam-Haroud, D. [1999], Path consistency on triangulated constraint graphs, *in* 'International Joint Conference on Artificial Intelligence (IJCAI 1999)', pp. 456–461.

[14] Bodirsky, M. and Dalmau, V. [2013], 'Datalog and constraint satisfaction with infinite templates', *Journal of Computer and System Sciences* **79**(1), 79–100.

[15] Bodirsky, M. and Wölfl, S. [2011], RCC8 is polynomial on networks of bounded treewidth, *in* 'International Joint Conference on Artificial Intelligence (IJCAI-11)', pp. 756–761.

[16] Caduff, D. and Egenhofer, M. J. [2007], 'Geo-mobile query-by-sketch', *International Journal of Web Engineering and Technology* **3**(2), 157–175.

[17] Chandra, P. and Pujari, A. K. [2005], Minimality and convexity properties in spatial CSPs, *in* 'International Conference on Tools for Artificial Intelligence (ICTAI 2005)', pp. 589–593.

[18] Chmeiss, A. and Condotta, J. [2011], Consistency of triangulated temporal qualitative constraint networks, *in* 'International Conference on Tools for Artificial Intelligence (ICTAI 2011)', pp. 799–802.

[19] Chmeiss, A., Krawczyk, V. and Sais, L. [2008], Redundancy in CSPs, *in* 'European Conference on Artificial Intelligence (ECAI 2008)', pp. 907–908.

[20] Clementini, E., Di Felice, P. and van Oosterom, P. [1993], A small set of formal topological relationships suitable for end-user interaction, *in* 'International Symposium on Spatial Databases', pp. 277–295.

[21] Clementini, E., Sharma, J. and Egenhofer, M. J. [1994], 'Modelling topological spatial relations: Strategies for query processing', *Computers & Graphics* **18**(6), 815 – 822.

[22] Cohen, D. A., Cooper, M. C., Creed, P., Marx, D. and Salamon, A. Z. [2012], 'The tractability of CSP classes defined by forbidden patterns', *Journal of Artificial Intelligence Research* **45**, 47–78.

[23] Cohn, A. G. and Gotts, N. M. [1996], 'The 'egg-yolk' representation of regions with indeterminate boundaries', *Geographic Objects with Indeterminate Boundaries* **2**, 171–187.

[24] Cohn, A. G. and Renz, J. [2008], Qualitative spatial representation and reasoning, *in* 'Handbook of Knowledge Representation', Elsevier, Amsterdam, Netherlands, pp. 551–596.

[25] Condotta, J.-F., Kaci, S. and Schwind, N. [2008], A framework for merging qualitative constraints networks, *in* 'The Twenty-First International FLAIRS Conference (FLAIRS 2008)', pp. 586–591.

[26] Danzer, L., Grünbaum, B. and Klee, V. [1963], Helly's theorem and its relatives, *in* 'Symposium in Pure Mathematics of the American Mathematical Society (Convexity)', pp. 101–179.

[27] De Berg, M., Van Kreveld, M., Overmars, M. and Schwarzkopf, O. C. [2000], *Computational Geometry: Algorithms and Applications*, Springer, Heidelberg, Germany.

[28] Dechter, R., Meiri, I. and Pearl, J. [1991], 'Temporal constraint networks', *Artificial Intelligence* **49**(1), 61–95.

[29] Deville, Y., Barette, O. and Van Hentenryck, P. [1999], 'Constraint satisfaction over connected row-convex constraints', *Artificial Intelligence* **109**(1), 243–271.

[30] Duckham, M., Kulik, L., Worboys, M. and Galton, A. [2008], 'Efficient generation of simple polygons for characterizing the shape of a set of points in the plane', *Pattern Recognition* **41**(10), 3224–3236.

[31] Düntsch, I. [2005], 'Relation algebras and their application in temporal and spatial reasoning', *Artificial Intelligence Review* **23**(4), 315–357.

[32] Düntsch, I., Wang, H. and McCloskey, S. [2001], 'A relation-algebraic approach to the region connection calculus', *Theoretic Computer Science* **255**(1-2), 63–83.

[33] Dylla, F., Lee, J. H., Mossakowski, T., Schneider, T., van Delden, A., van de Ven, J. and Wolter, D. [2016], 'A survey of qualitative spatial and temporal calculi - algebraic and computational properties', *CoRR* **abs/1606.00133**.

[34] Dylla, F., Mossakowski, T., Schneider, T. and Wolter, D. [2013], Algebraic properties of qualitative spatio-temporal calculi, *in* 'Conference on Spatial Information Theory (COSIT 2013)', pp. 516–536.

[35] Edelman, S. [1999], *Representation and recognition in vision*, MIT press Cambridge, MA.

[36] Edelsbrunner, H., Kirkpatrick, D. G. and Seidel, R. [1983], 'On the shape of a set of points in the plane', *IEEE Transactions on Information Theory* **29**(4), 551–559.

[37] Egenhofer, M. J. [1997], 'Query processing in spatial-query-by-sketch', *Journal of Visual Languages & Computing* **8**(4), 403–424.

[38] Egenhofer, M. J. and Franzosa, R. D. [1991], 'Point-set topological spatial relations', *International Journal of Geographical Information System* **5**(2), 161–174.

[39] Egenhofer, M. J. and Herring, J. [1991], Categorizing binary topological relations between regions, lines, and points in geographic databases, Technical report, Department of Surveying Engineering, University of Maine.

[40] Egenhofer, M. J. and Sharma, J. [1993], 'Assessing the consistency of complete and incomplete topological information', *Geographical Systems* **1**(1), 47–68.

[41] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. [1996], A density-based algorithm for discovering clusters in large spatial databases with noise, *in* 'ACM SIGKDD Conference on Knowledge Discovery and Data Mining', pp. 226–231.

[42] Finkel, R. A. and Bentley, J. L. [1974], 'Quad trees: A data structure for retrieval on composite keys', *Acta Informatica* **4**(1), 1–9.

[43] Fogliaroni, P. [2012], Qualitative spatial configuration queries – Towards next generation access methods for GIS, PhD thesis, University of Bremen, Germany.

[44] Fogliaroni, P., De Felice, G., Schmid, F. and Wallgrün, J. O. [2011], Managing qualitative spatial information to support query-by-sketch, *in* 'COSIT 2011 Workshop Understanding and Processing Sketch Maps', pp. 21–32.

[45] Frank, A. U. [1991], Qualitative spatial reasoning with cardinal directions, *in* 'Österreichische Artificial-Intelligence-Tagung / Austrian Conference on Artificial Intelligence (ÖGAI-91)', pp. 157–167.

[46] Freksa, C. [1992], 'Temporal reasoning based on semi-intervals', *Artificial Intelligence* **54**(1), 199–227.

[47] Freuder, E. C. [1978], 'Synthesizing constraint expressions', *Communications of the ACM* **21**(11), 958–966.

[48] Gerevini, A. [2005], 'Incremental qualitative temporal reasoning: Algorithms for the point algebra and the ORD-Horn class', *Artificial Intelligence* **166**(1-2), 37–80.

[49] Gerevini, A. E. and Saetti, A. [2011], 'Computing the minimal relations in point-based qualitative temporal reasoning through metagraph closure', *Artificial Intelligence* **175**(2), 556–585.

[50] Gerevini, A. and Schubert, L. [1995], 'On computing the minimal labels in time point algebra networks', *Computational Intelligence* **11**(3), 443–448.

[51] Gibson, D., Kumar, R. and Tomkins, A. [2005], Discovering large dense subgraphs in massive graphs, *in* 'International Conference on Very Large Data Bases (VLDB 2005)', pp. 721–732.

[52] Ginsberg, A. [1988], Knowledge-base reduction: A new approach to checking knowledge bases for inconsistency and redundancy, *in* 'The Seventh National Conference on Artificial Intelligence (AAAI 1988)', pp. 585–589.

[53] Glasgow, J. I., Fortier, S. and Allen, F. H. [1993], Molecular scene analysis: crystal structure determination through imagery, *in* 'Artificial Intelligence and Molecular Biology', pp. 433–458.

[54] Goodchild, M. F. [2007], 'Citizens as sensors: the world of volunteered geography', *GeoJournal* **69**(4), 211–221.

[55] Gottlob, G. and Fermüller, C. G. [1993], 'Removing redundancy from a clause', *Artificial Intelligence* **61**(2), 263 – 289.

[56] Goyal, R. K. and Egenhofer, M. J. [1997], The direction-relation matrix: A representation for directions relations between extended spatial objects, *in* 'The Annual Assembly and the Summer Retreat of University Consortium for Geographic Information Systems Science', pp. 22–81.

[57] Goyal, R. K. and Egenhofer, M. J. [2000], Consistent queries over cardinal directions across different levels of detail, *in* 'The 11th International Workshop on Database and Expert Systems Applications (DEXA 2000)', pp. 876–880.

[58] Grimm, S. and Wissmann, J. [2011], Elimination of redundancy in ontologies, *in* 'Extended Semantic Web Conference (ESWC 2011)', pp. 260–274.

[59] Güting, R. H. and Schilling, W. [1987], 'A practical divide-and-conquer algorithm for the rectangle intersection problem', *Information Sciences* **42**(2), 95–112.

[60] Güting, R. H. and Wood, D. [1984], 'Finding rectangle intersections by divide-and-conquer', *IEEE Transactions on Computers* **100**(7), 671–675.

[61] Guttman, A. [1984], R-trees: A dynamic index structure for spatial searching, *in* 'International Conference on Management of Data (ACM SIGMOD 1984)', pp. 47–57.

[62] Hoffart, J., Suchanek, F. M., Berberich, K. and Weikum, G. [2013], 'Yago2: A spatially and temporally enhanced knowledge base from wikipedia', *Artificial Intelligence* **194**, 28–61.

[63] Huang, J. [2012], Compactness and its implications for qualitative spatial and temporal reasoning, *in* 'International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)', pp. 500–508.

[64] Huang, J., Li, J. J. and Renz, J. [2013], 'Decomposition and tractability in qualitative spatial and temporal reasoning', *Artificial Intelligence* **195**, 140–164.

[65] Jonsson, P. and Drakengren, T. [1997], 'A complete classification of tractability in RCC-5', *Journal of Artificial Intelligence Research* **6**, 211–221.

[66] Kim, J., Vasardani, M. and Winter, S. [2015], 'From descriptions to depictions: A dynamic sketch map drawing strategy', *Spatial Cognition & Computation* **16**(1), 29–53.

[67] Knauff, M., Rauh, R. and Renz, J. [1997], A cognitive assessment of topological spatial relations: Results from an empirical investigation, *in* 'International Conference on Spatial Information (COSIT 1997)', pp. 193–206.

[68] Knauff, M., Rauh, R. and Schlieder, C. [1995], Preferred mental models in qualitative spatial reasoning: A cognitive assessment of allen's calculus, *in* 'The Seventeenth Annual Conference of the Cognitive Science Society', pp. 200–205.

[69] Knuth, D. E. [1968], 'The art of computer programming vol. 1, fundamental algorithms', *Addison-Wesley, Reading, MA* **9**, 364–369.

[70] Koubarakis, M., Kyzirakos, K., Karpathiotakis, M., Nikolaou, C., Sioutis, M., Vassos, S., Michail, D., Herekakis, T., Kontoes, C. and Papoutsis, I. [2011], Challenges for qualitative spatial reasoning in linked geospatial data, *in* 'IJCAI 2011 Workshop on Benchmarks and Applications of Spatial Reasoning (BASR-11)', pp. 33–38.

[71] Krokhin, A., Jeavons, P. and Jonsson, P. [2003], 'Reasoning about temporal relations: The tractable subalgebras of Allen's Interval Algebra', *Journal of the ACM* **50**(5), 591–640.

[72] Ladkin, P. B. and Maddux, R. D. [1988], On binary constraint networks, Technical report, Kestrel Institute, Palo Alto, California, USA.

[73] Lang, D., Winter, S. and Frank, A. U. [2001], 'Neighborhood relations between fields with applications to cellular networks', *GeoInformatica* **5**(2), 127–144.

[74] Leutenegger, S. T., Lopez, M. A. and Edgington, J. [1997], STR: A simple and efficient algorithm for R-tree packing, *in* B. Werner, ed., 'International Conference on Data Engineering (ICDE 1997)', IEEE Computer Society, USA, pp. 497–506.

[75] Li, J. J. and Li, S. [2013], On finding approximate solutions of qualitative constraint networks, *in* 'International Conference on Tools for Artificial Intelligence (ICTAI 2013)', pp. 30–37.

[76] Li, S. and Cohn, A. G. [2012], 'Reasoning with topological and directional spatial information', *Computational Intelligence* **28**(4), 579–616.

[77] Li, S., Long, Z., Liu, W., Duckham, M. and Both, A. [2015], 'On redundant topological constraints', *Artificial Intelligence* **225**, 51–78.

[78] Li, S. and Ying, M. [2003], 'Region Connection Calculus: Its models and composition table', *Artificial Intelligence* **145**(1-2), 121–146.

[79] Liberatore, P. [2005], 'Redundancy in logic I: CNF propositional formulae', *Artificial Intelligence* **163**(2), 203–232.

[80] Ligozat, G. [1993], Qualitative triangulation for spatial reasoning, *in* 'European Conference on Spatial Information Theory (COSIT 1993)', pp. 54–68.

[81] Ligozat, G. [1994], Tractable relations in temporal reasoning: pre-convex relations, *in* 'ECAI 1994 Workshop on Spatial and Temporal Reasoning', pp. 99–108.

[82] Ligozat, G. [1998], 'Reasoning about cardinal directions', *Journal of Visual Languages and Computing* **9**(1), 23–44.

[83] Ligozat, G. and Renz, J. [2004], What is a qualitative calculus? A general framework, *in* 'Pacific Rim International Conference on Artificial Intelligence', pp. 53–64.

[84] Liu, W. [2013], Qualitative constraint satisfaction problems: algorithms, computational complexity, and extended framework, PhD thesis, University of Technology Sydney, Australia.

[85] Liu, W. and Li, S. [2012], Solving minimal constraint networks in qualitative spatial and temporal reasoning, *in* 'International Conference on Principles and Practice of Constraint Programming (CP 2012)', pp. 464–479.

[86] Liu, W., Zhang, X., Li, S. and Ying, M. [2010], 'Reasoning about cardinal directions between extended objects', *Artificial Intelligence* **174**(12-13), 951–983.

[87] Long, Z., Sioutis, M. and Li, S. [2016], Efficient path consistency algorithm for large qualitative constraint networks, *in* 'International Joint Conference on Artificial Intelligence (IJCAI 2016)', pp. 1202–1208.

[88] Mark, D. M., Comas, D., Egenhofer, M. J., Freundschuh, S. M., Gould, M. D. and Nunes, J. [1995], Evaluating and refining computational models of spatial relations through cross-linguistic human-subjects testing, *in* 'International Conference on Spatial Information Theory (COSIT 1995)', pp. 553–568.

[89] Monk, J. D. and Bonnet, R. [1989], *Handbook of Boolean algebras*, Vol. 3, North Holland.

[90] Montanari, U. [1974], 'Networks of constraints: fundamental properties and applications to picture processing', *Information Sciences* **7**, 95–132.

[91] Mossakowski, T. and Moratz, R. [2012], 'Qualitative reasoning about relative direction of oriented points', *Artificial Intelligence* **180**, 34–45.

[92] Nebel, B. [1997], 'Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ORD-Horn class', *Constraints* **1**(3), 175–190.

[93] Nebel, B. and Bürckert, H.-J. [1995], 'Reasoning about temporal relations: A maximal tractable subclass of Allen's Interval Algebra', *Journal of the ACM* **42**(1), 43–66.

[94] Papadias, D. and Sellis, T. [1994], 'Qualitative representation of spatial knowledge in two-dimensional space', *The VLDB Journal* **3**(4), 479–516.

[95] Papadias, D., Theodoridis, Y., Sellis, T. K. and Egenhofer, M. J. [1995], Topological relations in the world of minimum bounding rectangles: A study with R-trees, *in* 'International Conference on Management of Data (ACM SIGMOD 1995)', pp. 92–103.

[96] Planken, L., de Weerdt, M. and van der Krogt, R. [2008], P^3C: A new algorithm for the Simple Temporal Problem, *in* 'International Conference on Automated Planning and Scheduling (ICAPS 2008)', pp. 256–263.

[97] Randell, D. A., Cui, Z. and Cohn, A. G. [1992], A spatial logic based on regions and connection, *in* 'International Conference on Principles of Knowledge Representation and Reasoning (KR 1992)', pp. 165–176.

[98] Renz, J. [1999], Maximal tractable fragments of the region connection calculus: A complete analysis, *in* 'International Joint Conference on Artificial Intelligence (IJCAI 1999)', pp. 448–455.

[99] Renz, J. and Ligozat, G. [2005], Weak composition for qualitative spatial and temporal reasoning, *in* 'International Conference on Principles and Practice of Constraint Programming (CP 2005)', pp. 534–548.

[100] Renz, J. and Nebel, B. [1999], 'On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus', *Artificial Intelligence* **108**(1), 69–123.

[101] Renz, J. and Nebel, B. [2001], 'Efficient methods for qualitative spatial reasoning', *Journal of Artificial Intelligence Research* **15**, 289–318.

[102] Renz, J. and Nebel, B. [2007], Qualitative spatial reasoning using constraint calculi, *in* 'Handbook of Spatial Logics', Springer, Heidelberg, Germany, pp. 161–215.

[103] Renz, J., Rauh, R. and Knauff, M. [2000], Towards cognitive adequacy of topological spatial relations, *in* 'Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications', pp. 184–197.

[104] Rigaux, P., Scholl, M. and Voisard, A. [2001], *Spatial databases: With application to GIS*, Morgan Kaufmann, Massachusetts, USA.

[105] Rodríguez, M. A., Egenhofer, M. J. and Blaser, A. D. [2003], Query preprocessing of topological constraints: Comparing a composition-based with neighborhood-based approach, *in* 'International Symposium on Spatial and Temporal Databases (SSTD 2001)', pp. 362–379.

[106] Schmolze, J. G. and Snyder, W. [1999], 'Detecting redundancy among production rules using term rewrite semantics', *Knowledge-Based Systems* **12**(1-2), 3–11.

[107] Schockaert, S., Smart, P. D., Abdelmoty, A. I. and Jones, C. B. [2008], Mining topological relations from the web, *in* 'International Conference on Database and Expert Systems Applications (DEXA 2008)', pp. 652–656.

[108] Scivos, A. and Nebel, B. [2004], The finest of its class: The natural point-based ternary calculus $\mathcal{LR}$ for qualitative spatial reasoning, *in* 'International Conference on Spatial Cognition (ICSC 2004)', pp. 283–303.

[109] Sioutis, M. and Condotta, J.-F. [2014*a*], Incrementally building partially path consistent qualitative constraint networks, *in* 'International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA 2014)', pp. 104–116.

[110] Sioutis, M. and Condotta, J.-F. [2014*b*], Vertex incremental path consistency for qualitative constraint networks, *in* 'Hellenic Conference on Artificial Intelligence (SETN 2014)', pp. 454–459.

[111] Sioutis, M., Condotta, J.-F. and Koubarakis, M. [2016], 'An efficient approach for tackling large real world qualitative spatial networks', *International Journal on Artificial Intelligence Tools* **25**(02), 1–33.

[112] Sioutis, M. and Koubarakis, M. [2012], Consistency of chordal RCC-8 networks, *in* 'International Conference on Tools for Artificial Intelligence (ICTAI 2012)', pp. 436–443.

[113] Sioutis, M., Li, S. and Condotta, J.-F. [2015], Efficiently characterizing non-redundant constraints in large real world qualitative spatial networks, *in* 'International Joint Conference on Artificial Intelligence (IJCAI 2015)', pp. 3229–3235.

[114] Sioutis, M., Long, Z. and Li, S. [2016], Efficiently reasoning about qualitative constraints through variable elimination, *in* 'Hellenic Conference on Artificial Intelligence (SETN 2016)', pp. 1–10.

[115] Sioutis, M., Salhi, Y. and Condotta, J.-F. [2015], A simple decomposition scheme for large real world qualitative constraint networks., *in* 'The Twenty-Eighth International FLAIRS Conference (FLAIRS 2015)', pp. 119–122.

[116] Skiadopoulos, S. and Koubarakis, M. [2005], 'On the consistency of cardinal direction constraints', *Artificial Intelligence* **163**(1), 91–135.

[117] Smith, T. R. and Park, K. K. [1992], 'Algebraic approach to spatial reasoning', *International Journal of Geographical Information Systems* **6**(3), 177–192.

[118] Stell, J. G. [2000], 'Boolean connection algebras: a new approach to the region-connection calculus', *Artificial Intelligence* **122**(1), 111–136.

[119] Tarjan, R. E. and Yannakakis, M. [1984], 'Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs', *SIAM Journal on Computing* **13**, 566–579.

[120] Valdés-Pérez, R. E. [1987], The satisfiability of temporal constraint networks., *in* 'The Sixth National Conference on Artificial Intelligence (AAAI 1987)', pp. 256–260.

[121] van Beek, P. [1992], 'Reasoning about qualitative temporal information', *Artificial Intelligence* **58**, 297–326.

[122] van Beek, P. and Cohen, R. [1990], 'Exact and approximate reasoning about temporal relations', *Computational Intelligence* **6**(3), 132–147.

[123] van Beek, P. and Dechter, R. [1995], 'On the minimality and global consistency of row-convex constraint networks', *Journal of the ACM* **42**, 543–561.

[124] Vasardani, M., Winter, S. and Richter, K.-F. [2013], 'Locating place names from place descriptions', *International Journal of Geographical Information Science* **27**(12), 2509–2532.

[125] Vilain, M. B. and Kautz, H. A. [1986], Constraint propagation algorithms for temporal reasoning, *in* 'The Fifth National Conference on Artificial Intelligence (AAAI 1986)', pp. 377–382.

[126] Vögele, T., Schlieder, C. and Visser, U. [2003], Intuitive modelling of place name regions for spatial information retrieval, *in* 'Conference on Spatial Information Theory (COSIT 2003)', Vol. 2825, pp. 239–252.

[127] Wallgrün, J. O. [2012], Exploiting qualitative spatial reasoning for topological adjustment of spatial data, *in* 'International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2012)', pp. 229–238.

[128] Wallgrün, J. O. [2013], Topological adjustment of polygonal data, *in* 'International Symposium on Spatial Data Handling (SDH 2013)', pp. 193–208.

[129] Wallgrün, J. O. and Dylla, F. [2010], 'Spatial data integration with qualitative integrity constraints', *Online Extended Abstract of International Conference on Geographic Information Science (GIScience 2010)* (Paper No. 142).

[130] Worboys, M. F. and Duckham, M. [2004], *GIS: A computing perspective*, CRC Press, Florida, USA.

[131] Zhang, X., Liu, W., Li, S. and Ying, M. [2008], Reasoning with cardinal directions: An efficient algorithm., *in* 'The Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 1988)', Vol. 8, pp. 387–392.

[132] Zhang, Y. and Freuder, E. C. [2008], 'Properties of tree convex constraints', *Artificial Intelligence* **172**(12-13), 1605–1612.

[133] Zhang, Y. and Marisetti, S. [2009], 'Solving connected row convex constraints by variable elimination', *Artificial Intelligence* **173**(12-13), 1204–1219.